



**Pedro Miguel  
Rocha Matos**

**Engenharia de Tráfego de Redes Ethernet  
baseadas em PBB-TE**



**Pedro Miguel  
Rocha Matos**

## **Engenharia de Tráfego de Redes Ethernet baseadas em PBB-TE**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestrado Integrado em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Professor Doutor Amaro de Sousa, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Professor Doutor Joaquim Madeira, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

## **o júri**

presidente

**Prof. Dr. Aníbal Manuel Oliveira Duarte**

professor catedrático do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

**Prof. Dr. Rui Jorge Morais Tomaz Valadas**

professor catedrático do Departamento de Engenharia Electrotécnica e de Computadores do Instituto Superior Técnico da Universidade Técnica de Lisboa

**Prof. Dr. Amaro Fernandes de Sousa (Orientador)**

professor auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

**Prof. Dr. Joaquim João Estrela Ribeiro Silvestre Madeira (Co-Orientador)**

professor auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

## **Agradecimentos**

Em primeiro lugar gostaria de agradecer ao Professor Amaro de Sousa e ao Professor Joaquim Madeira pela orientação e disponibilidade demonstrada ao longo de todo o projecto.

Gostaria de agradecer aos meus amigos, colegas de universidade, que foram um grande suporte em todo este caminho.

Uma palavra especial para o Tiago Condeixa, que me ajudou muito nestes anos finais do percurso académico.

Por último, mas não menos importante, à minha família pelo apoio dado e oportunidades fornecidas.

## Palavras-chave

Redes de telecomunicações, engenharia de tráfego, *Ethernet*, PBB-TE, protecção de tráfego.

## Resumo

Um dos aspectos mais importantes na engenharia de tráfego em redes de telecomunicações é a determinação do encaminhamento a adoptar para os fluxos de informação gerados pelos serviços suportados. Este encaminhamento deve ser criteriosamente escolhido para que os limites impostos pelas capacidades de transmissão das ligações da rede não sejam excedidos ou, caso contrário, criar-se-iam situações de congestão com a consequente perda de informação. Dada a importância crescente dos serviços de telecomunicações no dia-a-dia da sociedade actual, o encaminhamento, deve também garantir a protecção de tráfego, ou seja, que o tráfego é integralmente suportado mesmo quando há falhas na rede. Neste trabalho, considera-se o caso de uma única falha, quer seja de uma ligação ou de um *switch*.

A minimização das piores cargas de uma rede é um dos objectivos mais importantes da engenharia de tráfego, pois maximiza tanto a robustez da rede a variações não previsíveis de tráfego como a quantidade de tráfego adicional que é possível condicionar.

No entanto, é impossível determinar soluções óptimas, pois grande parte dos problemas de optimização em engenharia de tráfego é de complexidade exponencial. Assim, é necessário utilizar métodos heurísticos que encontram boas soluções, em tempos de cálculo úteis.

O *Provider Backbone Bridge – Traffic Engineering* (PBB-TE) é uma tecnologia recente proposta para redes *Ethernet*, que permite o fornecimento de serviços com engenharia de tráfego e qualidade de serviço, sendo adequado ao fornecimento de serviços de missão crítica nas redes de próxima geração. Permite a utilização de *switches Ethernet* de baixo custo em redes de operadores de larga escala, possibilitando que estes reduzam os seus custos de instalação e de operação.

O presente trabalho tem por objectivo o desenvolvimento de algoritmos eficientes, baseados em métodos heurísticos, de suporte às tarefas de engenharia de tráfego, referidas anteriormente, em redes *Ethernet* baseadas no PBB-TE.

**Keywords**

Telecommunications network, traffic engineering, Ethernet, PBB-TE, traffic protection

**Abstract**

One of the most important aspects of traffic engineering in telecommunications networks is the determination of the routing paths of the traffic flows generated by the supported services. These routing paths must be carefully determined so that the limits imposed by the capacity of network links are not exceeded; otherwise it would create situations of congestion with consequent loss of information. Given the growing importance of telecommunications services on our current society, the routing must also ensure traffic protection, i.e. all traffic must be fully supported even when there are failures in the network. In this work, is considered the case of a single failure, either in a link or in a switch.

The minimization of the worst loads of a network is one of the most important goals of traffic engineering; it maximizes both the robustness of the network to unpredictable changes of traffic and the amount of additional traffic that can be accommodated on the network.

However, it is impossible to determine optimal solutions, since most of these problems has exponential complexity. Therefore, it is necessary to resort to heuristic methods that are able to find good solutions in useful computational time.

The Provider Backbone Bridge - Traffic Engineering (PBB-TE) is a recent technology proposed for Ethernet networks, which allows the provision of services with traffic engineering and quality of service, being appropriate for delivering critical mission in next generation networks. It allows the use of low-cost Ethernet switches in the operator's large-scale networks, enabling them to reduce their costs of installation and operation.

The aim of this work is the development of efficient algorithms, based on heuristic methods, to support the tasks of traffic engineering, described above, in Ethernet networks based on PBB-TE.

## **Glossário de Acrônimos**

B-DA	Backbone DA
B-SA	Backbone SA
B-VID	Backbone VLAN ID
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
C-VID	Customer VLAN ID
DA	Customer Destination MAC
FSF	Forward Star Form
IEEE	Institute of Electrical and Electronics Engineers
I-SID	Service Identifier
LAN	Local Area Network
MAC	Media Access Control
MAN	Metropolitan Area Network
MPLS	Multi Protocol Label Switching
OSI	Open Systems Interconnection
PBB	Provider Backbone Bridges
PBB-TE	Provider Backbone Bridge - Traffic Engineering
PBT	Provider Backbone Transport
SA	Customer Source MAC
SDH	Synchronous Digital Hierarchy
SONET	Synchronous Optical NETworking
S-VID	Service VLAN ID
VLAN	Virtual LAN
VLAN ID	VLAN Identifier
WLAN	Wireless Local Area Network

## Índice

O júri

Agradecimentos

Resumo

Abstract

Glossário de Acrónimos

## Índice

Capítulo 1 – Introdução .....	1
1.1 – Motivação.....	1
1.2 – Objectivos .....	2
1.3 – Principais Contribuições .....	2
1.4 – Estrutura da Dissertação .....	2
Capítulo 2 – Tecnologias de Rede.....	5
2.1 – Introdução .....	5
2.2 – <i>Ethernet</i> .....	5
2.2.1 – Introdução .....	5
2.2.2 – História .....	6
2.2.3 – Características .....	7
2.2.4 – Versões .....	8
2.3 – Provider Backbone Bridges .....	10
2.3.1 – Introdução .....	10
2.3.2 – Evolução da <i>Ethernet</i> (LAN) até PBB (MAN) .....	10
2.4 – Provider Backbone Bridges - Traffic Engineering .....	11
2.4.1 – Características .....	11
2.4.2 – Modo de Funcionamento .....	12
Capítulo 3 – Algoritmos de Optimização para Redes de Telecomunicações .....	15
3.1 – Introdução .....	15
3.2 – Estratégia Básica de Todos os Algoritmos .....	17
3.3 – Algoritmo Construtivo .....	18
3.4 – Fases .....	19
3.4.1 – Introdução .....	19



3.4.2 – Fase 1 - Pesquisa Local Aleatória .....	19
3.4.3 – Fase 2 - Pesquisa por Pesos Modificados .....	22
3.4.4 – Fase 3 - Junção da Pesquisa Local Aleatória com Pesquisa por Pesos Modificados .....	25
3.5 – Principais Rotinas .....	26
3.5.1 – Sumário .....	26
3.5.2 – Cálculo dos Pesos para Determinação dos Pares de Percursos Disjuntos e Actualização da Rede .....	26
3.5.3 – Algoritmo de <i>Suurballe</i> .....	30
3.5.4 – Melhor Solução e Selecção do Tipo de Percurso .....	37
Capítulo 4 – Resultados Computacionais.....	39
4.1 – Sumário.....	39
4.2 – Casos de Estudo .....	39
4.3 – Resultados – Fases 1 e 2.....	41
4.4 – Resultados – Fase 3.....	55
Capítulo 5 – Conclusões .....	71
Apêndice A – Exemplo de Utilização .....	73
Apêndice B – Formato dos Dados e dos Resultados.....	77
Apêndice C – Resultados.....	79
Bibliografia .....	97

## Índice de Figuras

Figura 1. Rascunho da primeira rede <i>Ethernet</i> [10] .....	6
Figura 2. Fases do CSMA/CD [11].....	7
Figura 3. Trama <i>Ethernet</i> do PBB [4].....	11
Figura 4. Rede típica PBB-TE [16].....	13
Figura 5. Sequência do esquema do programa .....	16
Figura 6. Diagrama da Pesquisa Local Aleatória .....	21
Figura 7. Rede exemplo antes da modificação dos pesos.....	22
Figura 8. Rede exemplo depois da modificação dos pesos.....	23
Figura 9. Gráfico ilustrativo da variação geral do expoente .....	24
Figura 10. Gráfico ilustrativo da variação do expoente, para os valores representados... 25	
Figura 11. Exemplo de como se remove ligações comuns a dois percursos .....	31
Figura 12. Exemplo de um par de percursos disjuntos nas ligações.....	32
Figura 13. Exemplo de um par de percursos disjuntos nos nós .....	34
Figura 14. Rede constituída por 16 nós, 27 ligações .....	39
Figura 15. Rede constituída por 23 nós, 40 ligações .....	40
Figura 16. Rede constituída por 29 nós, 51 ligações .....	40
Figura 17. Espaço de soluções possíveis.....	66
Figura 18. Distribuição de Exp 5, Exp 10 e Exp 1000, para as duas redes de 23 nós. ....	67
Figura 19. Distribuição de Exp 5, Exp 10 e Exp 1000, para as duas redes de 29 nós. ....	68
Figura 20. Exemplo de um ficheiro de entrada.....	73
Figura 21. Rede esquematizada do exemplo anterior.....	73
Figura 22. Matriz com a capacidade de cada ligação do exemplo considerado .....	74
Figura 23. Exemplo de um ficheiro de saída .....	76
Figura 24. Formatação que deve ter o ficheiro de entrada .....	77
Figura 25. Configuração mínima que a rede pode apresentar.....	77
Figura 26. Formato do ficheiro de saída .....	78

## Índice de Tabelas

Tabela 1. Resultados relativos à rede 16, para pares de percursos disjuntos nas ligações – Fases 1 e 2.....	43
Tabela 2. Resultados relativos à rede 16, para pares de percursos disjuntos nos nós – Fases 1 e 2.....	44
Tabela 3. Resultados relativos à rede 23, para pares de percursos disjuntos nas ligações – Fases 1 e 2.....	45
Tabela 4. Resultados relativos à rede 23, para pares de percursos disjuntos nos nós – Fases 1 e 2.....	46
Tabela 5. Resultados relativos à rede 23A, para pares de percursos disjuntos nas ligações – Fases 1 e 2.....	47
Tabela 6. Resultados relativos à rede 23A, para pares de percursos disjuntos nos nós – Fases 1 e 2.....	48
Tabela 7. Resultados relativos à rede 29, para pares de percursos disjuntos nas ligações – Fases 1 e 2.....	49
Tabela 8. Resultados relativos à rede 29, para pares de percursos disjuntos nos nós – Fases 1 e 2.....	50
Tabela 9. Resultados relativos à rede 29A, para pares de percursos disjuntos nas ligações – Fases 1 e 2.....	51
Tabela 10. Resultados relativos à rede 29A, para pares de percursos disjuntos nos nós – Fases 1 e 2.....	52
Tabela 11. Opção 1 Vs Opção 4.....	53
Tabela 12. Resultados relativos à rede 23, para pares de percursos disjuntos nas ligações – Fase 3 .....	57
Tabela 13. Resultados relativos à rede 23, para pares de percursos disjuntos nos nós – Fase 3.....	58
Tabela 14. Resultados relativos à rede 23A, para pares de percursos disjuntos nas ligações – Fase 3 .....	59
Tabela 15. Resultados relativos à rede 23A, para pares de percursos disjuntos nos nós – Fase 3.....	60
Tabela 16. Resultados relativos à rede 29, para pares de percursos disjuntos nas ligações – Fase 3 .....	61
Tabela 17. Resultados relativos à rede 29, para pares de percursos disjuntos nos nós – Fase 3.....	62

Tabela 18. Resultados relativos à rede 29A, para pares de percursos disjuntos nas ligações – Fase 3 .....	63
Tabela 19. Resultados relativos à rede 29A, para pares de percursos disjuntos nos nós – Fase 3 .....	64
Tabela 20. Melhorias dos casos de estudo na fase 3, em relação às melhores soluções das fases anteriores. ....	65
Tabela 21. Resultados das redes de 29 nós, para pares de percursos disjuntos nas ligações, em relação às melhores soluções das fases anteriores. ....	65
Tabela 22. Resultados das 3 opções utilizadas para variação do expoente .....	68
Tabela 23. Lista dos arcos ordenados na Forward Star Form .....	74
Tabela 24. Ponteiro para cada nó na Forward Star Form .....	75
Tabela 25. Cargas dos arcos da solução inicial da rede 16, usando percursos disjuntos nas ligações e nos nós .....	80
Tabela 26. Cargas dos arcos da solução inicial da rede 23, usando percursos disjuntos nas ligações e nos nós .....	81
Tabela 27. Cargas dos arcos da solução inicial da rede 23A, usando percursos disjuntos nas ligações e nos nós .....	82
Tabela 28. Cargas dos arcos da solução inicial da rede 29, usando percursos disjuntos nas ligações e nos nós .....	84
Tabela 29. Cargas dos arcos da solução inicial da rede 29A, usando percursos disjuntos nas ligações e nos nós .....	85
Tabela 30. Estado das 16 piores cargas das soluções relativas à rede 16, para pares de percursos disjuntos nas ligações – Fases 1 e 2 .....	86
Tabela 31. Estado das 16 piores cargas das soluções relativas à rede 16, para pares de percursos disjuntos nos nós – Fases 1 e 2 .....	86
Tabela 32. Estado das 14 piores cargas das soluções relativas à rede 23, para pares de percursos disjuntos nas ligações – Fases 1 e 2 .....	87
Tabela 33. Estado das 14 piores cargas das soluções relativas à rede 23, para pares de percursos disjuntos nos nós – Fases 1 e 2 .....	87
Tabela 34. Estado das 13 piores cargas das soluções relativas à rede 23A, para pares de percursos disjuntos nas ligações – Fases 1 e 2 .....	88
Tabela 35. Estado das 13 piores cargas das soluções relativas à rede 23A, para pares de percursos disjuntos nos nós – Fases 1 e 2 .....	88
Tabela 36. Estado das 14 piores cargas das soluções relativas à rede 29, para pares de percursos disjuntos nas ligações – Fases 1 e 2 .....	89

Tabela 37. Estado das 14 piores cargas das soluções relativas à rede 29, para pares de percursos disjuntos nos nós – Fases 1 e 2 .....	89
Tabela 38. Estado das 14 piores cargas das soluções relativas à rede 29A, para pares de percursos disjuntos nas ligações – Fases 1 e 2 .....	90
Tabela 39. Estado das 14 piores cargas das soluções relativas à rede 29A, para pares de percursos disjuntos nos nós – Fases 1 e 2 .....	90
Tabela 40. Estado das 13 piores cargas das soluções relativas à rede 23, para pares de percursos disjuntos nas ligações – Fase 3 .....	91
Tabela 41. Estado das 13 piores cargas das soluções relativas à rede 23, para pares de percursos disjuntos nos nós – Fase 3 .....	92
Tabela 42. Estado das 13 piores cargas das soluções relativas à rede 23A, para pares de percursos disjuntos nas ligações – Fase 3 .....	92
Tabela 43. Estado das 13 piores cargas das soluções relativas à rede 23A, para pares de percursos disjuntos nos nós – Fase 3 .....	93
Tabela 44. Estado das 14 piores cargas das soluções relativas à rede 29, para pares de percursos disjuntos nas ligações – Fase 3 .....	94
Tabela 45. Estado das 14 piores cargas das soluções relativas à rede 29, para pares de percursos disjuntos nos nós – Fase 3 .....	94
Tabela 46. Estado das 14 piores cargas das soluções relativas à rede 29A, para pares de percursos disjuntos nas ligações – Fase 3 .....	95
Tabela 47. Estado das 14 piores cargas das soluções relativas à rede 29A, para pares de percursos disjuntos nos nós – Fase 3 .....	96

# Capítulo 1

## Introdução

### 1.1 – Motivação

A necessidade de reduzir custos, aumentar a confiabilidade e permitir a partilha de recursos físicos e informações provocou a criação das redes de computadores. Estas, ao apresentarem-se deste modo, continuam, hoje em dia, a evoluir progressivamente [1].

As redes informáticas e as tecnologias necessárias para a sua interligação continuam a liderar as indústrias de hardware, software e periféricos de computadores. O crescimento no número e nos tipos de utilizadores de redes são reflexos dessa expansão.

Actualmente, as redes de computadores estão no centro da comunicação moderna. Na década de 90, as telecomunicações tiveram um grande crescimento que não teria sido possível sem a evolução das redes de computadores.

A engenharia de tráfego é um método de optimização do desempenho de redes de telecomunicações que gere, analisa e prevê o comportamento dos dados transmitidos nestas.

A tecnologia de rede mais utilizada em todo o mundo é a *Ethernet* [2][3]. Esta foi melhorada ao longo dos anos e já se está a preparar o seu futuro. A combinação de tecnologias diferentes numa plataforma, como *Ethernet*, MPLS e mais recentemente PBB [4][5] (IEEE 802.1ah - *Provider Backbone Bridges*) e PBB-TE [6][7][8] (IEEE 802.1Qay - *Provider Backbone Bridge Traffic Engineering*) permitem que os *switches* sejam utilizados em redes de operadores de larga escala, possibilitando a redução dos seus custos de instalação e de operação. Estas tecnologias mais recentes, embora ainda não sejam suportadas pelos actuais *switches*, estão a ser propostas pelo IEEE (*Institute of Electrical and Electronics Engineers*) para uma melhor adequação da tecnologia *Ethernet* aos requisitos dos operadores públicos.

Assim, neste contexto, a motivação para o trabalho desenvolvido que serve de base a esta dissertação é optimizar o desempenho de uma rede de telecomunicações, tendo como principal consideração as características do PBB-TE.

## 1.2 – Objectivos

O principal objectivo desta dissertação é desenvolver e implementar um conjunto de algoritmos eficientes para engenharia de tráfego em redes de telecomunicações, cujos dados de entrada são os parâmetros da rede em estudo, enquanto os dados saída são relativos à melhor solução encontrada.

Para cumprir este objectivo, outros passos intermédios tiveram que ser atingidos:

- Estudar a tecnologia *Ethernet*;
- Estudar a tecnologia PBB-TE;
- Estudar técnicas de engenharia de tráfego;
- Estudar a eficiência de métodos de optimização;
- Desenvolver algoritmos eficientes baseados nas alíneas anteriores.

## 1.3 – Principais Contribuições

Como resultado do cumprimento dos objectivos propostos, este trabalho apresenta as seguintes contribuições:

- Implementação e desenvolvimento de algoritmos que determinam quais os melhores percursos para várias sequências de fluxos, minimizando as cargas máximas da rede;
- Propostas de técnicas heurísticas que procuram melhores soluções recorrendo a:
  - pesquisa local aleatória;
  - expoentes das cargas;
  - junção das técnicas anteriores.

## 1.4 – Estrutura da Dissertação

Este trabalho encontra-se dividido em cinco capítulos. Em seguida, será apresentado um breve resumo do conteúdo de cada capítulo.

No primeiro capítulo é feita uma introdução ao trabalho, enquadrando-o nas futuras redes *Ethernet*, indicando as motivações para a sua realização, assim como também os principais objectivos e contribuições. Apresenta-se ainda a estrutura da dissertação de modo a facilitar a sua consulta.

No segundo capítulo são descritas as tecnologias de rede subjacentes ao trabalho realizado nesta dissertação, apresentando a sua evolução e as principais características.

No terceiro capítulo é apresentado o desenvolvimento do trabalho, detalhando as suas fases e os algoritmos implementados que permitem otimizar uma rede de telecomunicações. Também são descritas outras implementações efectuadas e que ajudaram a tornar os algoritmos ainda mais eficientes.

O quarto capítulo diz respeito aos resultados computacionais, onde são testadas, para várias redes, as diversas fases dos algoritmos implementados, apresentando informações sobre as melhores soluções encontradas.

Finalmente, no quinto e último capítulo, são apresentadas as conclusões resultantes do trabalho efectuado e possível aplicação futura.

Para além destes capítulos, esta dissertação inclui os seguintes apêndices:

- Apêndice A: Exemplo de Utilização
- Apêndice B: Estado dos Dados
- Apêndice C: Resultados

No Apêndice A, é apresentado uma execução de um dos algoritmos implementados. Utiliza-se uma rede simples, com poucos nós, para uma melhor leitura dos dados relativos a ela.

No Apêndice B, são apresentados os parâmetros utilizados no ficheiro de entrada, que permite o correcto funcionamento dos algoritmos. É ainda descrito o conteúdo do ficheiro de saída.

No Apêndice C, encontram-se alguns resultados dos testes efectuados, que auxiliam a análise feita no Capítulo 4.



# Capítulo 2

## Tecnologias de Rede

### 2.1 – Introdução

Desde que surgiram as redes de computadores, estas não param de evoluir. Esta evolução é espelhada em termos de tecnologia, recursos físicos, quantidade de informação transmitida e até mesmo o número de utilizadores.

Hoje em dia, as redes informáticas são usadas em larga escala para diversos tipos de comunicação. A grande evolução que as telecomunicações sofreram, deveu-se, em parte, ao avanço progressivo destas redes.

Neste capítulo apresenta-se um resumo das tecnologias de rede que servem de base a este trabalho. Encontra-se dividido em 3 partes: *Ethernet*, *Provider Backbone Bridge* e *Provider Backbone Bridge - Traffic Engineering*.

### 2.2 – Ethernet

#### 2.2.1 – Introdução

A *Ethernet* é uma tecnologia de interligação para redes locais (*Local Area Networks – LAN*) e é baseada no envio de pacotes (pedaços de informação de tamanho variável). Esta tecnologia define a cablagem e os sinais eléctricos para a camada física, como também o formato de pacotes e de protocolos para a camada de controlo de acesso ao meio (*Media Access Control - MAC*) do modelo OSI (*Open Systems Interconnection*) [9].

Nesta secção é feita uma descrição geral da evolução da *Ethernet* ao longo dos anos, as suas características e as suas várias versões.

## 2.2.2 – História

A *Ethernet* foi inventada em 1973 por *Robert Metcalfe* no *Palo Alto Research Center* (PARC) da *Xerox*, e contribuiu durante os últimos 36 anos para a revolução das redes informáticas. Inicialmente, esta norma era chamada de “*Network Alto Aloha*”, sendo posteriormente modificada para *Ethernet*. A figura 1, apresenta o rascunho da primeira rede *Ethernet* projectada por *Robert Metcalfe*.

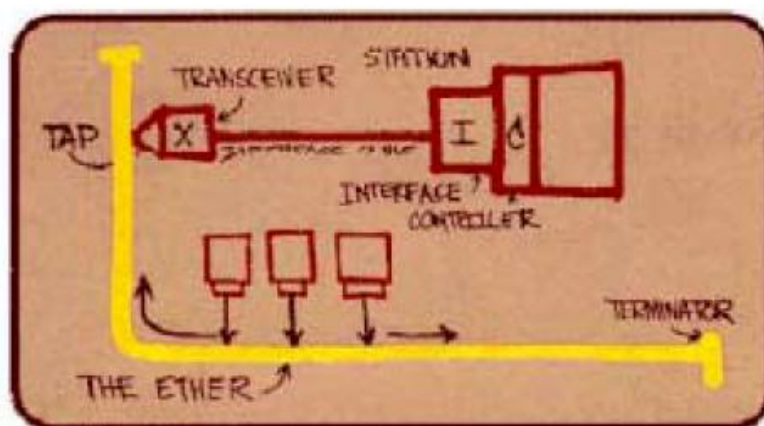


Figura 1. Rascunho da primeira rede *Ethernet* [10]

A falta de normalização dificultava o progresso das pesquisas e a venda de equipamentos. Assim, com o objectivo de tentar resolver este problema foi atribuída ao IEEE, em 1980, a responsabilidade de conceber e gerir a normalização da *Ethernet* - IEEE 802.3. Desde que foi regulamentada pelo IEEE as suas especificações foram totalmente disponibilizadas. Este aspecto conjugado com a facilidade de operação e com a sua robustez resultou no grande sucesso e utilização da *Ethernet* [1].

A *Ethernet* foi originalmente baseada na ideia de computadores comunicarem ao longo de um cabo coaxial partilhado agindo como um *broadcast transmission medium* (meio de transmissão em difusão). Os métodos utilizados revelam algumas semelhanças com sistemas de rádio, embora existam diferenças fundamentais, como o facto de ser muito mais fácil detectar colisões num cabo do que num sistema de rádio. Posteriormente, o cabo coaxial foi substituído por ligações ponto-a-ponto ligadas por *hubs* e/ou *switches Ethernet* para reduzir custos de instalação, aumentar a confiabilidade e permitir a gestão e resolução de problemas da transmissão ponto-a-ponto.

### 2.2.3 – Características

A *Ethernet* utiliza o CSMA/CD [3] (*Carrier Sense Multiple Access with Collision Detection*), que é um método de acesso que garante que apenas um nó da rede está a transmitir no cabo de rede, de cada vez, de modo a evitar transmissões simultâneas.

Este conceito pode ser dividido em:

- CS (*Carrier Sense*): Capacidade de identificar se está a ocorrer uma transmissão;
- MA (*Multiple Access*): Capacidade de múltiplos nós concorrerem pela utilização do cabo de rede;
- CD (*Collision Detection*): Responsabilidade em identificar colisões na rede.

Assim, o CSMA/CD processa-se em várias etapas (figura 2):

- Um terminal só envia informação para a rede apenas quando se verifica que não existe tráfego no cabo de rede;
- Cada terminal “escuta” a rede para detectar possíveis colisões;
- Se dois ou mais terminais enviam informação ao mesmo tempo – ocorre colisão;
- Os dois param de transmitir;
- Esperam um período de tempo aleatório para verificarem se podem retransmitir;
- A detecção da colisão funciona apenas a uma distância inferior a 2500 m.

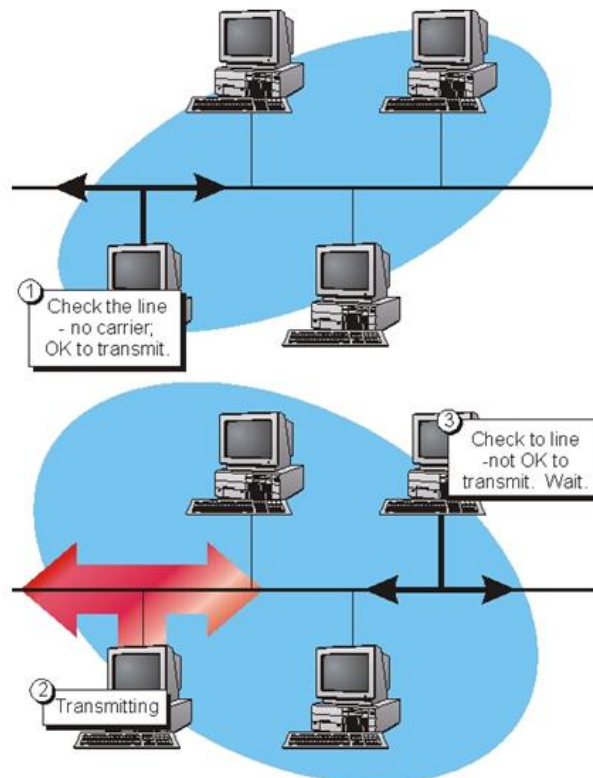


Figura 2. Fases do CSMA/CD [11]

Na *Ethernet* é forçado um tamanho mínimo de 64 bytes que conjugado com a distância máxima entre quaisquer duas estações garante a detecção de colisões. Também é imposto um intervalo mínimo entre o fim de uma transmissão e o início de nova transmissão (IFS - *Inter Frame Spacing*) para as estações terem tempo de receber uma nova trama. Quando o meio é detectado ocupado as estações continuam a escutar até que o meio seja detectado livre. Quanto isto ocorre, transmitem imediatamente (o protocolo diz-se 1-persistente). Depois de detectarem uma colisão, as estações esperam um tempo aleatório até retransmitir, definido por um algoritmo de recuo binário exponencial truncado [12].

Algumas tecnologias sem fios utilizam a *Ethernet* como rede de interligação (por exemplo a WLAN – IEEE 802.11 [13]). Neste caso, a *Ethernet* auxilia na interligação de diferentes células dessas redes sem fios.

## 2.2.4 – Versões

Em geral, a maioria das diferenças entre as várias versões de *Ethernet* podem ser parametrizadas em velocidade e cablagem. Em termos de *software*, a maioria das versões apresentadas em seguida, têm funcionamento idêntico entre si [3].

As primeiras versões de *Ethernet* foram a 10Base5, a 10Base2 e a 10BaseT. Estas suportavam taxas de transmissão de 10 Mb/s.

A 10Base5 foi a primeira tecnologia *Ethernet*. Faz parte da especificação de camada física de banda de base IEEE 802.3 e suporta uma taxa de transmissão de 10 Mb/s num máximo de 500 m de comprimento dos segmentos coaxiais. Também conhecida como *Thicknet*, usa um cabo coaxial grosso de 50 Ohm (*thick Ethernet*) ao qual são ligadas as estações através de um *transceiver*. A interface entre as estações e o *transceiver* é a AUI (*Attachment Unit Interface*) e o seu comprimento máximo é de 50 m. As distâncias entre *transceivers* têm de ser múltiplas de 2.5 m para minimizar reflexões. Por segmento, podem existir até 100 estações e estes podem ser interligados através de repetidores. Contudo, os sinais não podem viajar por mais de 5 segmentos e 4 repetidores.

A 10Base2, também conhecida como *thinnet* ou *cheapnet*, utiliza cabo coaxial fino de 50 Ohm (*thin Ethernet*). O cabo transmite sinais a 10 Mb/s e a uma distância máxima de 185 m, por segmento. As estações são ligadas ao cabo coaxial directamente através de um conector BNC e no máximo pode haver 30 estações por segmento, sendo que a distância mínima entre estações é de 0.5 m. Os segmentos podem ser interligados através de repetidores.

No sistema 10BaseT, cada estação liga-se directamente a um repetidor/*hub*, através de um cabo formado por 2 pares entrançados não-blindados (cabo UTP - *Unshielded Twisted Pair*) com conectores RJ-45. O comprimento máximo destes cabos é de 100 m e, no máximo, pode haver 4 *hubs* entre estações.

Posteriormente, criou-se a *Fast Ethernet*. Este termo designa várias normas da *Ethernet* que transportam o tráfego de dados à taxa nominal de 100 Mb/s, contra a taxa de transmissão de 10 Mb/s da *Ethernet* original, referida anteriormente, tornando assim a *Ethernet* 10 vezes mais rápida. Para efectivar esta alteração, as alternativas seriam aumentar o limite mínimo do comprimento do pacote em 10 vezes, diminuir em 10 vezes a extensão máxima da rede ou então fazer uma combinação das duas situações anteriores. A opção tomada para as normas de 100 Mb/s do IEEE 802.3 foi a diminuição da extensão máxima da rede. Outro facto relevante é que se permite a ocorrência colisões sem serem detectadas.

A *Fast Ethernet* tem implementações em cobre e fibra óptica. A implementação em cobre é dividida em:

- 100BASE-TX
- 100BASE-T4
- 100BASE-T2

A implementação em fibra óptica é dividida em:

- 100BASE-FX
- 100BASE-SX
- 100BASE-BX

Recentemente, surgiu a *Gigabit Ethernet*, que apresenta taxas de transmissão na ordem dos 1000 Mb/s (1 Gb/s) e em comparação à *Fast Ethernet*, optou-se por, adicionalmente, estender o comprimento mínimo do pacote para 512 octetos. Assim, os pacotes mais pequenos sofrem uma extensão de portadora, havendo múltiplos pequenos pacotes para transmitir, estes podem ser transmitidos *costas-com-costas* (o primeiro pacote tem de ser estendido para 512 octetos).

A *10-Gigabit Ethernet* é a norma mais recente da *Ethernet* e a mais rápida em uso, sendo 10 vezes mais rápida que a *Gigabit Ethernet*, transmitindo a 10 Gb/s.

## 2.3 – Provider Backbone Bridges

### 2.3.1 – Introdução

O *Provider Backbone Bridges* [4][5] (PBB - IEEE 802.1ah também conhecido por Mac-in-Mac ou MinM) é um conjunto de arquiteturas e protocolos de *routing* da rede do cliente e da rede do operador que permite a interligação de múltiplas redes *Provider Bridge* sem perder as VLANs definidas individualmente para cada cliente.

O PBB permite uma muito maior escalabilidade, porque proporciona fisicamente mais espaço de endereçamento na trama *Ethernet*. Além disso, reduz a quantidade de estado da rede necessária para manter e aumentar a segurança através da separação clara do espaço de endereçamento do cliente e do operador.

### 2.3.2 – Evolução da *Ethernet* (LAN) até PBB (MAN)

Inicialmente a *Ethernet* foi definida como uma tecnologia LAN (*Local Area Network*) para interligar os computadores, que estavam próximos uns dos outros, dentro de pequenas organizações.

Com o passar dos anos, a *Ethernet* tornou-se uma tecnologia tão popular que fez com que passasse a ser o mecanismo padrão da camada de ligação de dados (camada 2 do modelo OSI) [14] para transporte de dados. Como tal, houve a necessidade de estender a *Ethernet* desde a LAN até à MAN (*Metropolitan Area Network*), ou seja, alargar a área destas redes até à de uma cidade, por exemplo. O serviço de *Ethernet* numa escala MAN também é conhecido como *Provider bridging domain*.

Assim, o *Provider Bridges* [4] (IEEE 802.1ad, QinQ ou *Q-tunneling*) resulta da adição do *service provider* VLAN ID (S-VID) à trama *Ethernet* do cliente. Este novo campo é usado para identificar o serviço na rede do operador enquanto que o VLAN ID do cliente (C-VID) se mantém inalterado.

O *Provider Bridges* não oferece uma completa separação dos domínios do cliente e do operador, no entanto supera as limitações do espaço do VLAN ID. Contudo, não oferece a escalabilidade requerida pelos serviços *Ethernet* em escalas metropolitanas. Quando combinado com outros recursos, como, por exemplo, *control protocol tunneling* ou *Spanning Tree*, o *Provider Bridges* ajuda na separação dos domínios de controlo do cliente e do operador. Tem ainda o problema de ter muito pouco controlo sobre os endereços MAC, porque o encaminhamento continua a ser baseado nos endereços destino do cliente. Por isso, há a necessidade de ter melhores mecanismos.

E assim surge o PBB, oferecendo uma clara separação dos domínios do cliente e do operador. O PBB expande a trama *Ethernet* adicionando um cabeçalho MAC dedicado ao *service provider* e, ao fazê-lo, acrescenta à trama *Ethernet* do cliente um endereço MAC do *Backbone* origem (B-SA), um endereço MAC do *Backbone* destino (B-DA), um *Backbone* VLAN ID (B-VID) e um *Backbone Service ID* (I-SID) - figura 3.

Payload	
Ethertype	
C-VID	C-VID = Customer VLAN ID
Ethertype	
S-VID	S-VID = Service VLAN ID
Ethertype	
SA	SA = Customer Source MAC
DA	DA = Customer Destination MAC
I-SID	I-SID = Service ID
Ethertype	
B-VID	B-VID = Backbone VLAN ID
Ethertype	
B-SA	B-SA = Backbone SA
B-DA	B-DA = Backbone DA

Figura 3. Trama *Ethernet* do PBB [4]

O campo *Ethertype* define qual o protocolo de camada superior que o pacote transporta.

## 2.4 – Provider Backbone Bridges - Traffic Engineering

### 2.4.1 – Características

O *Provider Backbone Bridge - Traffic Engineering* [6][7][8] (PBB-TE - IEEE 802.1Qay) foi desenvolvido em 2006, sendo resultado da normalização do *Provider Backbone Transport* (PBT) da *Nortel* [15].

O PBB-TE é uma tecnologia inovadora que permite o fornecimento determinístico de serviços com engenharia de tráfego, qualidade de serviço e recursos de OAM (Operação, Administração, Manutenção). Baseia-se nos recursos da *Ethernet*, tornando-se assim o mais adequado no fornecimento de serviços de missão crítica nas redes de próxima geração.

Com esta tecnologia é agora possível construir redes totalmente baseadas em tecnologia *Ethernet* sem a necessidade de redes de apoio para garantir a qualidade *class-carrier* (como SDH/SONET ou MPLS) [7].

O PBB-TE reutiliza implementações existentes de VLANs e dupla marcação, e combina-os com a separação da rede e princípios de *layering* do PBB. No entanto, não usa mecanismos de *flooding/broadcasting* nem protocolos *Spanning Tree* - estes são desactivados por definição - e, portanto, pretende-se utilizá-lo na ligação orientada para aplicações de rede. Ao contrário do PBB, o PBB-TE não cria dinamicamente as suas tabelas de encaminhamento, pois estas são configuradas estaticamente.

Comparando o PBB-TE, ao PBB e aos seus predecessores, pode-se dizer que tem um comportamento mais previsível e que pode ser mais facilmente controlado pelo operador de rede, à custa de se exigir uma ligação de configuração em cada *bridge/switch* ao longo de uma via de transmissão.

Uma ligação *Ethernet* com PBB-TE permite o desenvolvimento de uma infra-estrutura *Ethernet* que suporta todas as características ópticas clássicas de infra-estruturas – como 50ms de *protection switching*, transparência de protocolo, qualidade de serviço e uma gestão de falhas robusta em elementos de *software*. Estas características permitiram que os prestadores de serviços fizessem com que o negócio de redes fosse rentável e escalável.

O PBB-TE é uma tecnologia ideal para a construção da infra-estrutura da *Ethernet*, porque melhora não só os níveis de desempenho da ligação, mas fá-lo de uma forma que é totalmente coerente com o simples circuito centrado operacional utilizado na agregação de metro e em redes de transportes [16].

## 2.4.2 – Modo de Funcionamento

No PBB-TE, cada serviço é identificado por um I-SID (*Backbone Service Instance Identifier*) e é associado a um tronco PBB-TE (caminhos ponto-a-ponto com engenharia de tráfego). Cada tronco PBB-TE é identificado pelos seguintes campos:

- B-SA (*Backbone – Source Address*) - identifica a *bridge* origem;
- B-DA (*Backbone - Destination Address*) - identifica a *bridge* destino;
- B-VID (*Backbone – VLAN ID*) - identifica a VLAN do *backbone*.

O B-SA e o B-DA são os terminais de cada tronco PBB-TE. O campo B-VID é utilizado para distinguir os diferentes troncos que possuem o mesmo destino.

O sistema de gestão configura os troncos PBB-TE em todas as *bridges*, tanto no extremo como no *core* da rede, através da criação de entradas na base de dados de encaminhamento estático. Este sistema é também responsável por garantir que não haja *loops* no encaminhamento dos pacotes.

As *bridges* situadas nos extremos dos *backbones* desempenham funções de encapsulamento/dencapsulamento do cabeçalho MAC e as do *core* actuam como nós de trânsito. Os pacotes são encaminhados segundo o VLAN ID exterior (B-VID) e o endereço MAC destino (B-DA).

Como referido atrás, o encaminhamento de pacote é baseado em entradas de base de dados de encaminhamento estático, não usando assim a aprendizagem dinâmica dos endereços MAC. Qualquer *frame broadcast* ou *multicast* é descartada ou encapsulada como *unicast* dentro do tronco. Todos os pacotes *Destination Lookup Failure* (DLF) são ignorados em vez de serem *flooded*. Com a eliminação de *broadcasting* e de *flooding*, e utilizando apenas o encaminhamento livre de *loops*, não será preciso usar o protocolo *Spanning Tree*.



Um dos aspectos mais importantes do PBB-TE é o *protection switching*. Em cada I-SID podem ser configurados dois percursos, um de serviço e outro de protecção, que são identificados pelo B-VID respectivo, como indica a figura 4. O *protection switching* ocorre quando o percurso de serviço falha. Neste instante, a *bridge* da origem troca (*switch*) o valor do B-VID para redireccionar o fluxo de tráfego para o percurso de protecção pré-definido, demorando cerca de 50 ms.

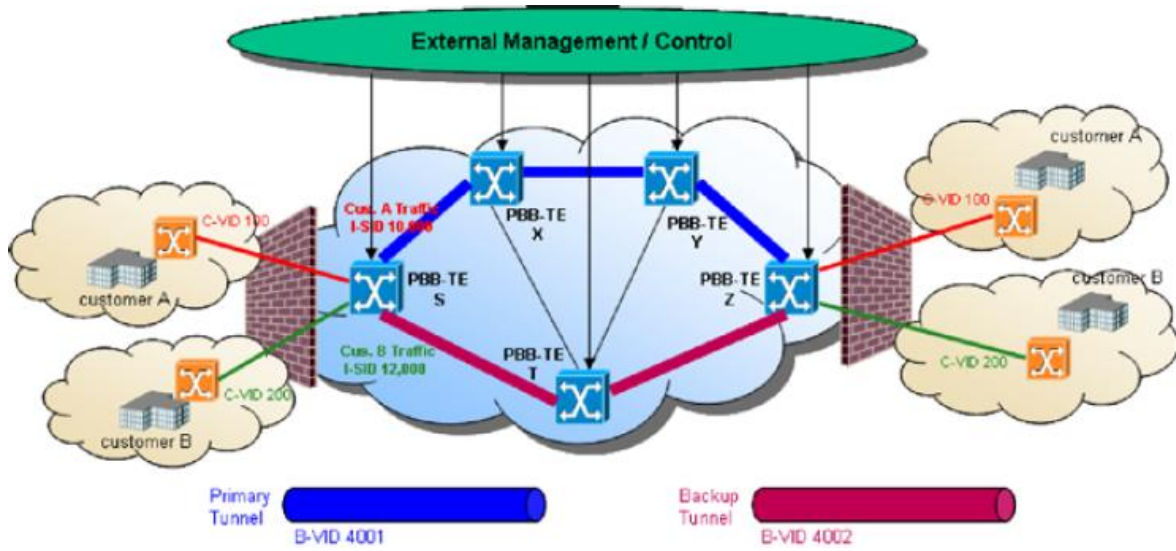


Figura 4. Rede típica PBB-TE [16]

Na figura 4, apresenta-se uma rede PBB-TE onde estão configurados o percurso de serviço, *Primary Tunnel*, e o percurso de protecção, *Backup Tunnel*, identificados, cada um, pelo respectivo B-VID.

Os equipamentos do PBB-TE impulsionam as economias de escala inerentes à *Ethernet*, prometendo soluções que são 30% a 40% mais baratas que as redes T-MPLS (*Transport - Multi Protocol Label Switching*) com características e capacidades semelhantes. Assim, o PBB-TE consegue um melhor retorno sobre o investimento efectuado [8].

## Capítulo 3

# Algoritmos de Optimização para Redes de Telecomunicações

### 3.1 – Introdução

No futuro, prevê-se que o uso da Internet seja ainda mais generalizado do que é actualmente e que todo o tráfego de rede seja protegido contra falhas. Nessa perspectiva, o PBB-TE tem vindo a ganhar força visto ser adaptado para ambientes MAN e possibilitar o encaminhamento estático com protecção [6].

No Capítulo 2, caracterizou-se o PBB-TE e referiu-se o *protection switching* como um aspecto fundamental desta tecnologia. Como dito anteriormente, o PBB-TE permite a existência de dois percursos configurados, um de serviço e um de protecção, sendo que o *protection switching* ocorre quando o percurso de serviço falha e o fluxo de tráfego passa a ser encaminhado pelo percurso de protecção. Estes percursos têm que ser disjuntos para que uma falha num deles não afecte o outro.

O objectivo deste trabalho é a determinação da melhor sequência destes pares de percursos (de serviço e de protecção), tentando reduzir as cargas mais elevadas da rede, isto é, reduzindo a taxa de utilização de cada uma das ligações. Esta minimização de cargas é um dos objectivos mais importantes da engenharia de tráfego, pois torna a rede mais robusta a variações inesperadas de tráfego e aumenta a quantidade de tráfego adicional que é possível condicionar.

Cada rede utilizada, ao longo deste trabalho, é composta por *switches* (nós/vértices) conectados por ligações ponto-a-ponto. A rede é representada pelo grafo  $G = (N, A)$ , em que  $N$  é o conjunto de nós,  $A$  é o conjunto das ligações entre os nós em que cada ligação  $\{i, j\} \in A$  tem uma capacidade de largura de banda. Os nós são *full-duplex*, isto é, permitem transferência de dados em ambos sentidos (1 ligação = 2 arcos de sentidos opostos,  $(i, j)$  e  $(j, i)$ ). A rede suporta um conjunto de VLANs. Cada VLAN suporta um fluxo de tráfego bi-direccional caracterizado por vértice origem, vértice destino, largura de banda da origem para o destino ( $\langle LB \text{ origem} \rightarrow \text{destino} \rangle$ ) e do destino para a origem ( $\langle LB \text{ destino} \rightarrow \text{origem} \rangle$ ). A cada arco está associada uma carga, que é dada pelo quociente

entre a largura de banda utilizada pelos fluxos que passam na ligação e a capacidade da ligação. Quando a carga for utilizada para efeitos de cálculo será denominada por peso.

Partindo dos pressupostos anteriormente apresentados, foram implementados diferentes algoritmos que têm como objectivo determinar, para cada fluxo, um par de percursos de encaminhamento, de modo que o conjunto final desses percursos minimize as piores cargas da rede, ou seja, as cargas mais elevadas nos arcos da rede.

Genericamente, estes algoritmos, que foram implementados em linguagem C, lêem um ficheiro de entrada com os dados relativos à topologia de uma rede e informação dos fluxos para os quais se pretende determinar os pares de percursos disjuntos, e escreve num ficheiro de texto os resultados finais obtidos (figura 5). Este último ficheiro contém o estado de cada arco (carga) e a sequência dos fluxos/percursos encontrada para a melhor solução. Considera-se como melhor solução, aquela que apresentar as piores cargas mais baixas. O Apêndice A apresenta um exemplo de utilização dos algoritmos implementados e o Apêndice B descreve como devem estar organizados os parâmetros no ficheiro de entrada e como se apresenta o ficheiro de saída. Os algoritmos são executados enquanto o número limite de iterações não for atingido, considerando que uma iteração corresponde a uma tentativa de determinação dos pares de percursos disjuntos do conjunto de fluxos. Na iteração zero (0), um algoritmo construtivo gera uma solução inicial e nas iterações seguintes tenta-se melhorar essa “melhor solução”, sendo que no final a melhor solução encontrada é escrita no ficheiro de saída.

Neste capítulo vão ser descritos os aspectos fundamentais dos algoritmos implementados, começando pelas descrições de uma estratégia básica e do algoritmo construtivo comuns a todos os algoritmos, depois são explicadas as três variantes/fases do trabalho e finalmente são detalhados os blocos constituintes mais relevantes dos algoritmos.

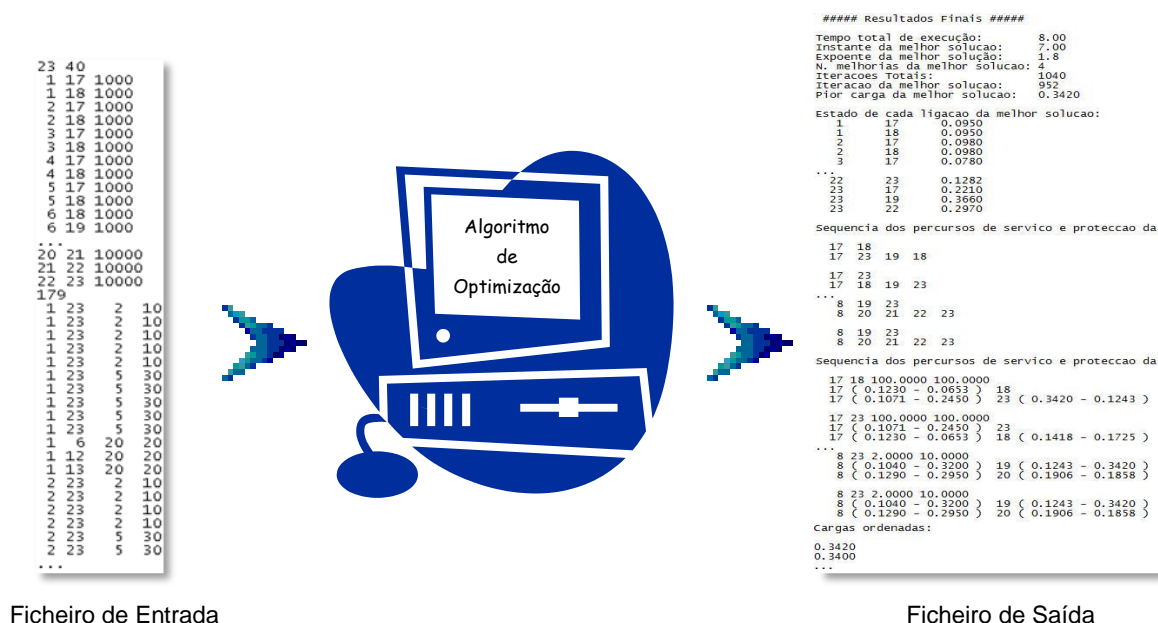


Figura 5. Sequência do esquema do programa

## 3.2 – Estratégia Básica de Todos os Algoritmos

Todos os algoritmos implementados usam a estratégia seguinte como forma de obter uma solução a partir de uma ordenação dos fluxos suportados pela rede:

1	Ordenação de fluxos
2	Todas as cargas da rede = 0
3	Para cada fluxo, de acordo com a ordem dos fluxos, é determinado 1 par de percursos disjuntos:
3.1	Cálculo dos pesos para determinação dos pares percursos disjuntos
3.2	Algoritmo de <i>Suurballe</i> (determinação dos pares percursos disjuntos)
3.3	<b>if</b> todas as cargas do par de percursos disjuntos são menores ou iguais a 100% <b>do:</b>
3.4	Actualizar as ligações por onde passam os percursos
3.5	<b>else</b>
3.6	Interromper sequência de fluxos
4	Retorna a solução gerada

Independentemente do algoritmo utilizado, para uma dada ordenação de fluxos (passo 1), as cargas de todos os arcos da rede são inicializadas a zero. Em seguida, são processados, por ordem, todos os fluxos dessa ordenação, determinando, para cada fluxo, um par de percursos disjuntos (passo 3) – este passo é analisado ao pormenor no subcapítulo 3.5.2. Antes de se iniciar a determinação do par de percursos disjuntos, para cada fluxo, é necessário calcular os pesos dos arcos, caso os percursos a determinar passem nestes arcos (passo 3.1). Depois, recorre-se ao algoritmo de *Suurballe* [17] para determinar o par de percursos disjuntos do fluxo que está a ser processado (passo 3.2) – este passo é pormenorizado no subcapítulo 3.5.3. Depois da determinação destes percursos, é preciso verificar se estes são válidos, isto é, se todas as cargas dos arcos pertencentes a estes percursos são menores ou iguais a 100% (passo 3.3). Caso esta condição seja satisfeita, então actualizam-se as cargas das ligações por onde passam esses percursos (passo 3.4) – este passo é explicado no subcapítulo 3.5.2 – e repetem-se os passos 3.1 a 3.4, para os outros fluxos da ordenação. Caso contrário (passo 3.5), interrompe-se a determinação desta ordenação de fluxos (passo 3.6), não tendo sido possível encontrar uma solução completa para a ordem dos fluxos considerada no início. No final, é devolvida a solução encontrada (passo 4). Esta solução é válida se foram determinados todos os pares de percursos disjuntos da ordenação de fluxos analisada e se estes forem válidos. Caso a determinação dos pares de percursos disjuntos tenha sido cancelada, então a solução originada é inválida.

Para implementar um algoritmo baseado em pesquisa local por troca de fluxos foi necessário alterar o passo 1 (ordenação dos fluxos) e para um algoritmo baseado em pesquisa por modificação de pesos alterou-se o passo 3.1 (cálculo dos pesos para determinação dos pares percursos disjuntos). Ambas as técnicas de pesquisa são descritas mais à frente no trabalho.

### 3.3 – Algoritmo Construtivo

O primeiro passo de todos os algoritmos implementados ao longo deste trabalho é a execução de um algoritmo construtivo, sendo que a solução, por ele gerada, é atribuída como melhor solução inicial para futuras comparações. Este algoritmo tem os seguintes passos:

- 1 Ordenação inicial dos fluxos
- 2 Todas as cargas da rede = 0
- 3 Para cada fluxo, de acordo com a ordem dos fluxos, é determinado 1 par de percursos disjuntos
- 4 Retorna a solução gerada

Antes de processar a ordenação inicial de fluxos, esta é ordenada de modo decrescente, segundo as respectivas larguras de banda. Comparam-se as larguras de bandas mais elevadas de cada fluxo, caso estas sejam iguais comparam-se as outras larguras de banda.

Considerando as seguintes larguras de bandas da ordenação inicial de fluxos:

Fluxo 1: 10 20  
Fluxo 2: 30 10  
Fluxo 3: 20 20  
Fluxo 4: 10 40

Os fluxos ficariam ordenados da seguinte maneira:

Fluxo 4: 10 40  
Fluxo 2: 30 10  
Fluxo 3: 20 20  
Fluxo 1: 10 20

Com esta ordenação consegue-se fazer com que os fluxos com maiores larguras de banda sejam processados em primeiro lugar, de modo a otimizar a carga da rede, permitindo, assim, que sejam determinados os pares de percursos para todos os fluxos (solução válida). Esta ordenação inicial não foi a primeira a ser testada. Fizeram-se alguns testes com ordenações aleatórias iniciais e, em grande parte dos testes realizados, originavam piores soluções que a ordenação inicial utilizada.

Como dito no subcapítulo 3.2, inicialmente as cargas de todos os arcos da rede estão a 0% e são alteradas sempre que é determinado um par de percursos disjuntos. Assim, pela ordem inicialmente estabelecida, vai sendo determinado sequencialmente um par de percursos disjuntos, para cada fluxo.

No final, a solução gerada é constituída pelo conjunto das cargas dos arcos da rede que é resultado da determinação dos pares de percursos disjuntos de todos os fluxos.

O algoritmo construtivo corresponde à primeira execução da estratégia apresentada no subcapítulo 3.2.

## 3.4 – Fases

### 3.4.1 – Introdução

Para se obter um algoritmo que melhore a solução obtida, pelo algoritmo construtivo inicial, foi necessário passar por diversas alternativas. Primeiro, começou-se por implementar um algoritmo, baseado na alteração da ordenação de fluxos, trocando a cada iteração um par de fluxos. Com este método procura-se encontrar a ordenação de fluxos que origina a melhor solução encontrada, pois ordens diferentes de fluxos, geralmente, resultam em soluções diferentes.

Posteriormente, foi implementado outro algoritmo com base na modificação dos pesos a serem utilizados na determinação dos pares de percursos disjuntos. Nesta técnica a ordenação dos fluxos é sempre a mesma (ordenação inicial) e procura-se melhorar as soluções através de um equilibrar das cargas da rede.

Finalmente, conjugaram-se estas duas hipóteses, num terceiro algoritmo, com o objectivo de melhorar ainda mais as soluções obtidas em cada um dos algoritmos anteriores.

Em seguida, vão ser apresentados estes três algoritmos, correspondendo, cada um, a uma das fases do trabalho realizado.

### 3.4.2 – Fase 1 - Pesquisa Local Aleatória

Na primeira fase do trabalho, o algoritmo concebido baseia-se na ideia seguinte: a cada iteração, antes da determinação dos pares de percursos disjuntos de uma dada ordenação de fluxos, troca-se aleatoriamente um par de fluxos, desde que estes não sejam iguais, isto é, tenham a mesma origem, destino e correspondentes larguras de banda. Se os índices dos fluxos aleatoriamente obtidos, corresponderem a fluxos iguais, gera-se outro índice dum fluxo para ser trocada a sua posição.

Em seguida, procede-se à determinação dos pares de percursos disjuntos, baseada em percursos de peso mínimo, segundo a ordem de fluxos que poderá resultar numa solução. Sempre que esta solução “aleatória” (solução actual) é gerada vai ser comparada com a melhor solução encontrada até ao momento – esta comparação encontra-se descrita com maior pormenor no subcapítulo 3.5.4. Se a solução actual for melhor que a “melhor solução”, então os dados relativos da solução actual são copiados para a melhor solução. Se a solução actual for pior que a melhor solução, então repõe-se a ordenação dos fluxos pela qual estavam ordenados na iteração anterior. Se a solução actual for igual à melhor solução, então esta não vai ser atribuída à melhor solução, nem se repõe a ordem dos fluxos, ou seja, o próximo passo será trocar novo par de fluxos e repetir os procedimentos anteriores. Os procedimentos a adoptar para estas três últimas condições foram determinados depois de vários testes computacionais, onde foram evidentes as melhorias nas soluções resultantes dos algoritmos que tinham estes procedimentos implementados. Parte destes testes podem ser consultados no Capítulo 4.

Depois da comparação com a melhor solução, repetem-se todos os passos anteriores, fazendo uma nova pesquisa local, por troca de pares de fluxos, com o objectivo de tentar melhorar a “melhor solução”, diferindo a ordenação actual da anterior num par de fluxos. Quando é atingido o número limite de iterações, vão ser nomeados, para cada par de percursos disjuntos, um percurso de serviço e um percurso de protecção, como vai ser descrito no subcapítulo 3.5.4.

O algoritmo relativo à fase 1 pode ser descrito pelos seguintes passos e pelo diagrama presente na figura 6:

1	Algoritmo Construtivo
2	Atribuição da solução inicial como melhor solução
3	<b>while</b> número de iterações não é atingido <b>do</b> :
4	troca aleatória de 1 par de fluxos
5	todas as cargas da rede = 0
6	determinação dos pares de percursos disjuntos para a sequência de fluxos
7	<b>if</b> solução actual for válida <b>do</b> :
8	comparação com a melhor solução
9	<b>if</b> solução actual melhor que “melhor solução” <b>do</b> :
10	melhor solução = solução actual
11	<b>else</b> solução actual pior que “melhor solução” <b>do</b> :
12	repor a ordem de fluxos em vigor antes da última troca
13	<b>else</b>
14	repor a ordem de fluxos em vigor antes da última troca
15	Seleccção dos percursos de serviço e de protecção

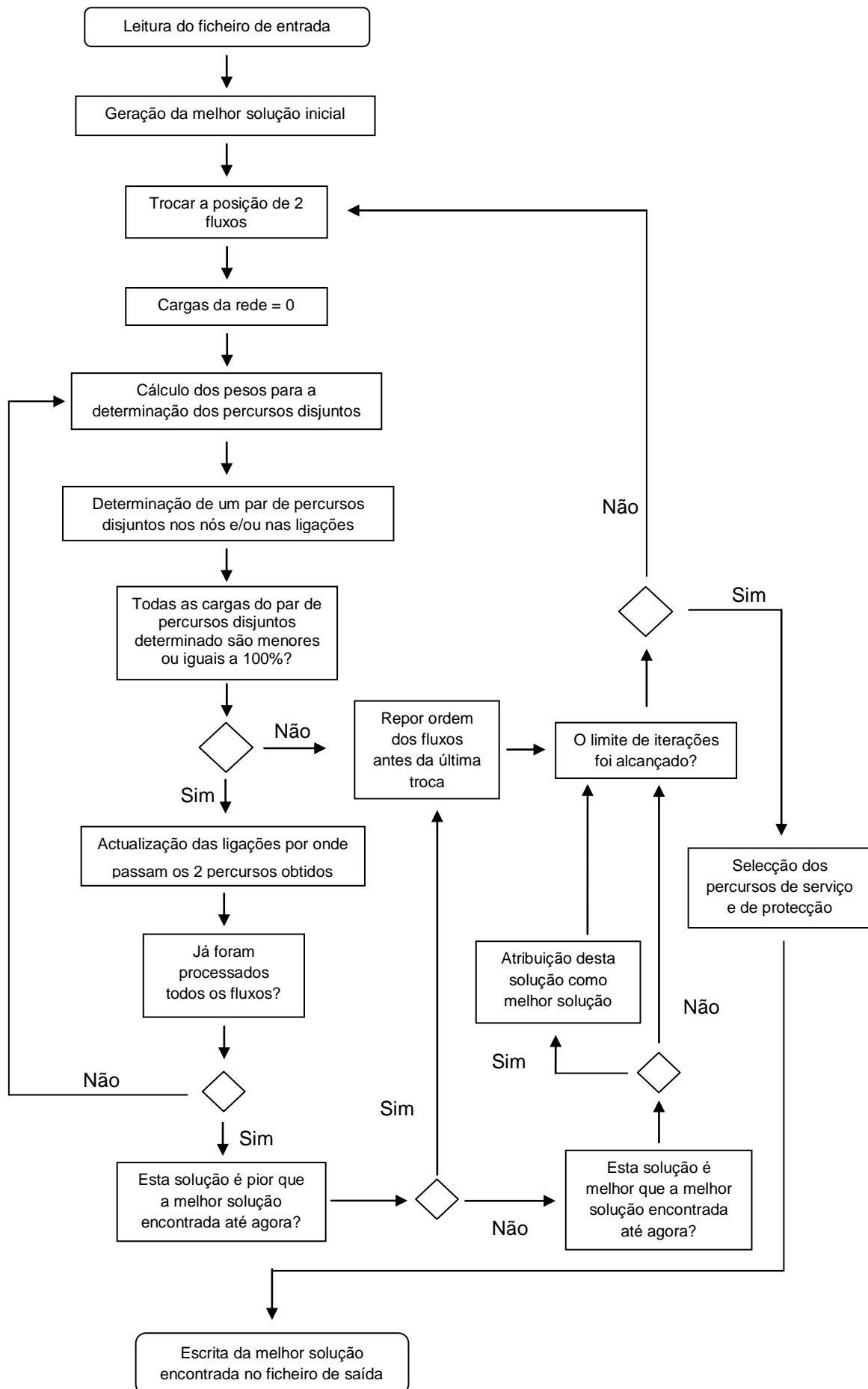


Figura 6. Diagrama da Pesquisa Local Aleatória



### 3.4.3 – Fase 2 - Pesquisa por Pesos Modificados

O algoritmo desenvolvido nesta fase segue uma estratégia um pouco diferente da anterior. Para além de não se efectuar nenhuma troca de fluxos (não se processa o passo 4 da Pesquisa Local Aleatória), logo a ordem destes não é alterada (não se processa os passos 11 a 14 da Pesquisa Local Aleatória) e efectua-se um procedimento extra na determinação dos pares de percursos disjuntos (passo 6 da Pesquisa Local Aleatória).

Para melhor compreensão da estratégia utilizada neste algoritmo, considere-se o seguinte exemplo, em que os pesos têm valores entre 0 e 1:

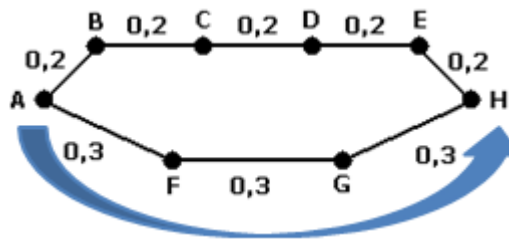


Figura 7. Rede exemplo antes da modificação dos pesos

Observando a figura 7, facilmente constata-se que o percurso de peso mínimo de A para H é A-F-G-H (peso total = 0.9), pois o caminho A-B-C-D-E-H tem peso total de 1.0. O que se pretende fazer é que o percurso seleccionado seja aquele que tenha as piores cargas mais baixas, ou seja, passar de A-F-G-H (piores cargas – 0.3, 0.3) para A-B-C-D-E-H (piores cargas - 0.2, 0.2).

Para que a selecção do melhor percurso fosse feita, de acordo com as piores cargas mais baixas, o método utilizado foi elevar todos os pesos da rede a um expoente. Como os pesos dos arcos variam de 0 a 1, e como o algoritmo utilizado para determinar os pares de percursos disjuntos (algoritmo de *Suurballe*) é baseado nos pesos totais dos percursos, então ao elevar os pesos a um expoente, faz com que as diferenças entre os pesos dos arcos seja mais acentuada, reflectindo-se no peso total dos percursos. Por exemplo, o peso de um arco é 0.3 e de outro é 0.9. Este último é três vezes maior que o primeiro. Elevando ambos pesos ao expoente 2, fica-se com 0.09 e 0.81, respectivamente. Ou seja, o arco com peso 0.9 passou de três vezes para nove vezes mais carregado que o arco com peso 0.3. É com base nestas diferenças, que são determinados percursos alternativos, não respeitando os percursos de peso mínimo, mas sim, pelos percursos com as piores caras mais baixas. É importante referir que estas alterações efectuadas nos pesos da rede só são utilizadas na determinação dos pares de percursos disjuntos (algoritmo de *Suurballe*), logo não afectam directamente as cargas da rede.

Voltando ao exemplo da figura 7 e elevando todos os seus pesos ao expoente 2, obtém-se:

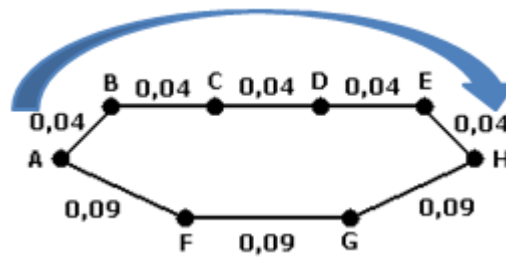


Figura 8. Rede exemplo depois da modificação dos pesos

Observando a figura 8, depois de alterar os pesos das ligações, verifica-se que o percurso de peso mínimo de A para H é agora A-B-C-D-E-H (peso total = 0.2), enquanto que o percurso A-F-G-H tem um peso total de 0.27 ( $> 0.2$ ), conseguindo deste modo fazer com que o percurso inicial com as piores cargas mais baixas (0.2) seja escolhido.

Deste modo, o algoritmo relativo à fase 2 pode ser descrito pelos seguintes passos:

1	Algoritmo Construtivo
2	Atribuição da solução inicial como melhor solução
3	<b>while</b> número de iterações não é atingido <b>do</b> :
4	todas as cargas da rede = 0
5	determinação dos pares de percursos disjuntos para a sequência de fluxos (com alteração de pesos)
6	<b>if</b> solução actual for válida <b>do</b> :
7	comparação com a melhor solução
8	<b>if</b> solução actual melhor que “melhor solução” <b>do</b> :
9	melhor solução = solução actual
10	Seleccção dos percursos de serviço e de protecção

Em termos de implementação, esta alteração reflecte-se nos procedimentos presentes no subcapítulo 3.5.2, antes do passo 11 é necessário modificar os pesos das ligações. Assim, a cada peso da rede é aplicado um expoente de valor variável. Na primeira iteração o expoente apresenta valor 1, sendo que nas restantes iterações é adicionado um parâmetro fixo *delta*, previamente escolhido pelo utilizador. Depois, continua-se o processo de determinação dos pares de percursos disjuntos, explicado no subcapítulo 3.5.2. Considera-se que uma iteração é uma tentativa de determinação dos pares de percursos disjuntos do conjunto de fluxos, por isso, quando não é gerada uma solução válida ou quando foi determinado o par de percursos disjuntos do último fluxo da ordenação, é incrementado o número de iterações, soma-se *delta* ao valor do expoente da iteração anterior e repete-se a determinação dos pares de percursos disjuntos do conjunto de fluxos.

Se para 1000 iterações, for considerado um *delta* de 0.1, então a gama dos expoentes testados será [1;100]. Para o mesmo número de iterações, caso seja utilizado um *delta* de 0.5, então a gama de expoentes é [1;500] e assim sucessivamente.

Assim, ao contrário do que se sucedia na fase 1, a técnica utilizada na fase 2 privilegia os percursos com as piores cargas mais baixas em detrimento dos percursos de peso mínimo total. Contudo, esta técnica depende da diferença entre as piores cargas dos percursos existentes entre o mesmo par origem-destino, pois quanto maior é essa diferença, maior será o valor do expoente necessário para que o percurso com as piores cargas mais baixas seja seleccionado.

Esta técnica procura minimizar as cargas mais elevadas da rede, evitando que certas ligações fiquem mais sobrecarregadas que outras, aumentando assim o número de percursos possíveis a adoptar pelos fluxos.

Em seguida, mostra-se o efeito que o expoente provoca nos pesos originais (figura 9) e pequenos exemplos concretos (figura 10):

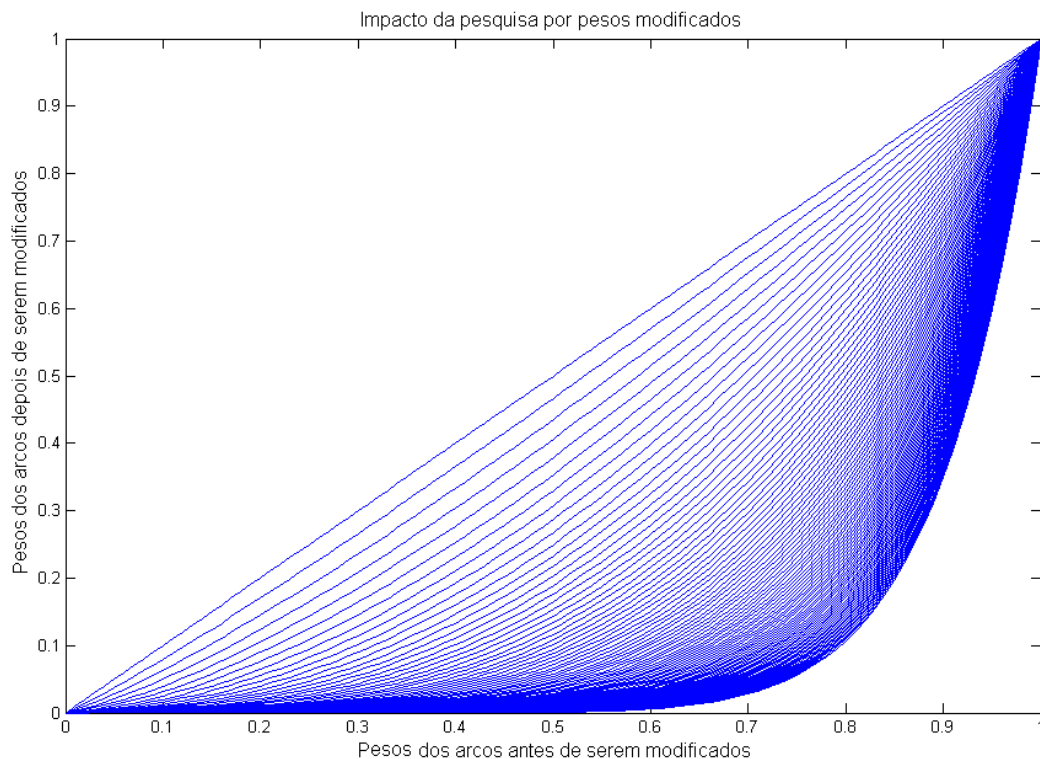


Figura 9. Gráfico ilustrativo da variação geral do expoente

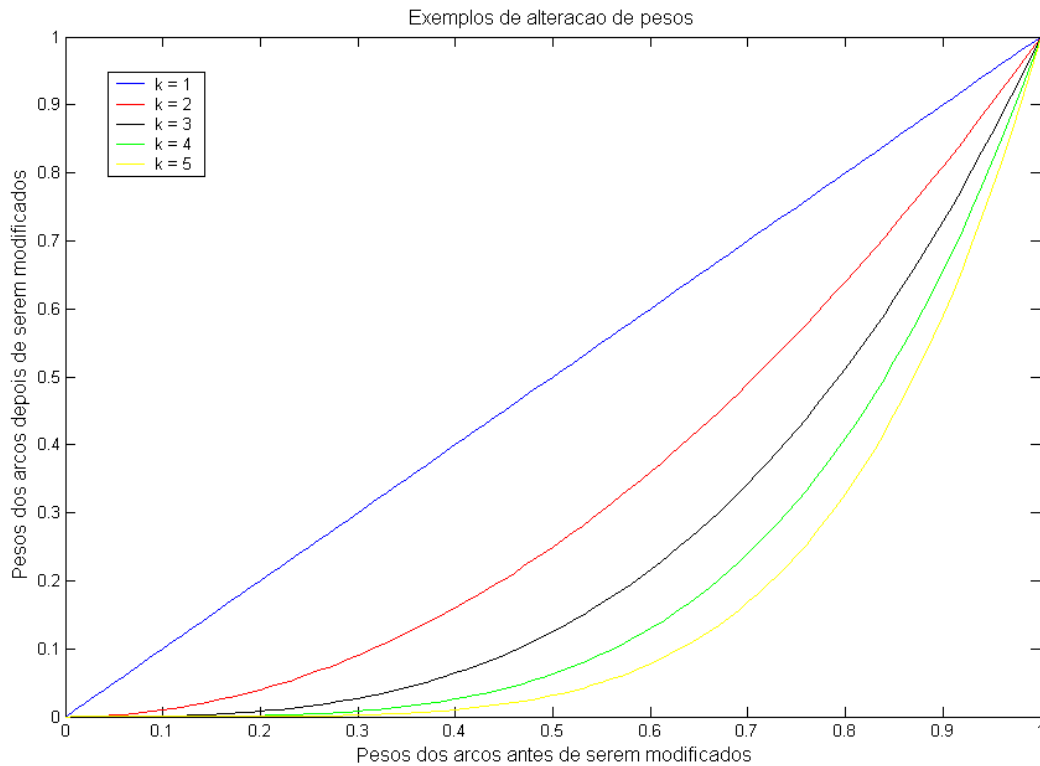


Figura 10. Gráfico ilustrativo da variação do expoente, para os valores representados

Vale a pena referir que estas alterações efectuadas nos valores dos pesos não prejudicam significativamente o desempenho dos algoritmos.

### 3.4.4 – Fase 3 - Junção da Pesquisa Local Aleatória com Pesquisa por Pesos Modificados

Finalmente, na terceira e última fase do trabalho, combinaram-se os algoritmos desenvolvidos nas fases anteriores, aliando assim a pesquisa local aleatória por troca de fluxos com a pesquisa por modificação dos pesos.

Os passos desta fase podem ser seguidos desta maneira:

- |   |  |
|---|--|
| 1 | Algoritmo Construtivo  |
| 2 | Atribuição da solução inicial como melhor solução local            |
| 3 | Pesquisa por Pesos Modificados                                     |
| 4 | Atribuição da melhor solução local como melhor solução global      |
| 5 | <b>while</b> número limite de iterações não é atingido <b>do</b> : |
| 6 | Pesquisa Local Aleatória   |
| 7 | Pesquisa por Pesos Modificados                                     |
| 8 | Comparação da melhor solução local com a melhor solução global     |
| 9 | Seleccção dos percursos de serviço e de protecção                  |

De referir que os valores de expoentes e de *delta* utilizados para a Pesquisa por Pesos Modificados nesta fase, são apresentados no Capítulo 4, assim como também os testes realizados que os justificam.

Nesta fase, tem-se dois tipos de “melhor solução”: Melhor Solução Local e Melhor Solução Global. A primeira é usada, para cada ordenação de fluxos, à medida que a Pesquisa por Pesos Modificados é executada, isto é, enquanto se vai variando os valores do expoente a aplicar aos pesos. As soluções aqui geradas são comparadas com a Melhor Solução Local. Terminada a Pesquisa por Pesos Modificados, a Melhor Solução Global é comparada com a Melhor Solução Local. Tal como acontece na primeira vez que esta pesquisa é executada, quando a Melhor Solução Local é “melhor” que a Melhor Solução Global, para esta última é copiada a Melhor Solução Local.

## 3.5 – Principais Rotinas

### 3.5.1 – Sumário

Em seguida, vão descritas as principais rotinas comuns a todas as fases do trabalho:

- Cálculo dos Pesos para Determinação dos Pares de Percursos Disjuntos;
- Actualização da Rede;
- Algoritmo de *Suurballe*;
- Comparação da Melhor Solução;
- Selecção do Tipo de Percurso.

### 3.5.2 – Cálculo dos Pesos para Determinação dos Pares de Percursos Disjuntos e Actualização da Rede

Um dos aspectos mais importantes dos algoritmos é a determinação do par de percursos de cada fluxo. Para isso, usa-se o algoritmo de *Suurballe* [17] que, dado um peso não negativo de cada arco, determina o par de percursos disjuntos de peso mínimo. Como esta operação é feita sequencialmente para cada fluxo, quando se processa um fluxo, usa-se o peso de cada arco resultante dos percursos dos fluxos anteriormente escolhidos e adiciona-se a este valor o pior peso que resultará na respectiva ligação (a que cada arco pertence) se o fluxo for encaminhado pelo arco no sentido da origem para o destino.

Essencialmente, calcula-se o estado futuro de todos os arcos da rede, caso os percursos determinados passem por lá. Depois, o peso dos arcos resultante destes cálculos serve de base ao algoritmo de *Suurballe* para calcular um par de percursos disjuntos com a mesma origem e o mesmo destino do fluxo. Caso estes percursos sejam válidos, as ligações por onde eles passem são actualizadas e repete-se o processo

descrito para o fluxo seguinte. Caso contr rio, termina-se a determina  o dos pares de percursos para esta ordena  o de fluxos.

Considerando as seguintes vari veis:

$N$  – n mero de v rtices da rede  
 $NF$  – n mero de fluxos  
 $A$  – conjunto de arcos da rede  
 $LB1_j$  – largura de banda da origem para o destino do fluxo  $j$   
 $LB2_j$  – largura de banda do destino para a origem do fluxo  $j$   
 $cap_{ij}$  – capacidade do arco  $ij$   
 $w_{ij}$  – peso do arco  $ij$   
 $antesW_{ij}$  – c pia de  $w_{ij}$  antes de ser alterado

Os passos relativos   determina  o dos pares de percursos disjuntos s o:

```

1  while  $j \leq NF$  do:
2      for  $(i, j) \in A$  do:
3           $antesW_{ij} = w_{ij}$ 
4      for  $(i, j) \in A$  do:
5           $aux1 = LB1_j / cap_{ij} + antesW_{ij}$ 
6           $aux2 = LB2_j / cap_{ji} + antesW_{ji}$ 
7          if  $\max(aux1, aux2) > 1$  do:
8               $w_{ij} = N$ 
9          else
10              $w_{ij} = \max(aux1, aux2)$ 
11         Invocar o Algoritmo de Suurballe
12         if todas as cargas do par de percursos disjuntos s o menores ou
            iguais a 100% do:
13             Actualizar as liga  es por onde passam os percursos
14         else
15             Interromper sequ ncia de fluxos

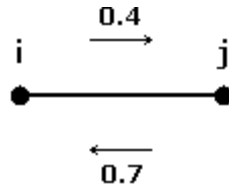
```

Cada fluxo   tratado sequencialmente pela ordem em vigor (passo 1). N o esquecer que quando esta fun  o   chamada pela primeira vez, os fluxos est o dispostos por ordem decrescente, de acordos com as larguras de banda de cada um, como explicado no subcap tulo 3.3.

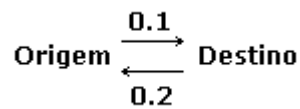
Come a-se por calcular o peso com que ficaria cada arco, caso o fluxo a ser processado passe por ele (passo 4 a 10). Este valor   obtido pela soma do peso actual do arco com o quociente entre a largura de banda do fluxo e a capacidade da liga  o (passo 5). No entanto, n o se sabe qual ser  o sentido que o fluxo ir  tomar, por isso, tem-se que considerar em cada arco, ambas larguras de banda,  $\langle LB \text{ origem} \rightarrow \text{destino} \rangle$   $\langle LB \text{ destino} \rightarrow \text{origem} \rangle$  (passo 6). Isto  , para a liga  o  $\{i, j\}$  n o se sabe se o sentido

directo do fluxo será pelo arco  $(i,j)$  ou pelo arco  $(j,i)$ . Por isso, o cálculo descrito anteriormente tem que ser feito para ambos os sentidos, considerando as respectivas larguras de banda.

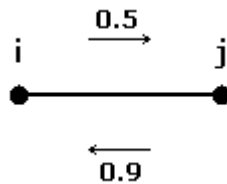
Considere o seguinte exemplo com os pesos de cada arco da ligação  $\{i,j\}$ :



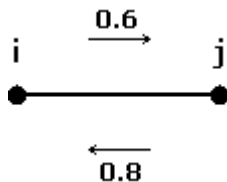
Pretende-se encaminhar o fluxo com os seguintes pesos:



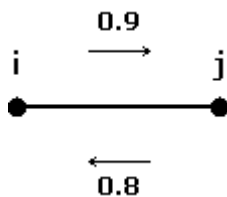
Se o fluxo passar na ligação  $\{i,j\}$  pelo arco  $(i,j)$ , o estado da ligação (i.e. peso da ligação) fica em cada um dos sentidos:



Se o fluxo passar na ligação  $\{i,j\}$  pelo arco  $(j,i)$ , o estado da ligação (i.e. peso da ligação) fica em cada um dos sentidos:



Assim, atribuindo a cada arco o pior dos casos, resulta em:



No caso de algum destes valores ser maior que 1 (100%) - passo 7, ou seja, o arco passaria a ser usado acima das suas possibilidades, é atribuído valor igual ao número de vértices da rede (passo 8). Ao atribuir este valor, garante-se que o fluxo não passa nesse arco, pois numa rede, o peso total máximo que um percurso pode ter é igual ao número de vértices menos um. Este é o caso extremo em que um percurso passa por todos os vértices da rede e os arcos por onde passa estão todos a 100%.

No caso de ambos os valores serem menores ou iguais a 1 (passo 9), ou seja, o arco tem largura de banda suficiente para o fluxo ser transmitido através dele, atribui-se a este arco o pior caso dos pesos obtidos (passo 10). Ou seja, após ter sido efectuado o cálculo para ambos os sentidos da ligação, o peso com que o arco ficará é o maior obtido, independentemente da sua orientação.

Os cálculos descritos acima são repetidos para todos os arcos da rede, completando assim o vector com as projecções dos pesos de todas as ligações, considerando que os possíveis percursos, para o fluxo em questão, passem por cada arco.

Neste ponto é usado um dos parâmetros de entrada introduzidos, permitindo assim a selecção do tipo de algoritmo de *Suurballe* que se pretende utilizar (passo 11) – determinar pares de percursos disjuntos nas ligações ou pares de percursos disjuntos nos nós. Imediatamente, executa-se este algoritmo, cujos procedimentos encontram-se no subcapítulo 3.5.3.

Nesta fase, independentemente do algoritmo de *Suurballe* utilizado, já se sabe qual o par de percursos disjuntos para o fluxo que está a ser processado. No entanto, antes de actualizar a rede com estes percursos é preciso validá-los, isto é, as cargas dos arcos que pertencem às ligações do par de percursos têm que ser menores ou iguais a 100%. De facto, o que realmente se faz é verificar se o peso total de cada percurso, no sentido directo e no sentido inverso, é menor que o número de vértices da rede, pois se esse número for igualado ou superado significa que pelo menos um arco tem carga superior a 100%. Assim, depois de verificar esta condição necessária a ambos percursos pode-se continuar ou interromper o processamento da ordenação de fluxos.

Caso ambos os percursos sejam válidos (passo 12), actualizam-se as ligações por onde eles passam (passo 13): ao peso de cada ligação pertencente aos caminhos obtidos vão ser somados os pesos dados por este fluxo. Assim, e utilizando as larguras de banda de cada fluxo presentes no ficheiro de entrada, <LB origem->destino> <LB destino->origem>, o peso a ser somado ao peso de um arco presente num caminho é dada pelo quociente de <LB origem->destino> com a capacidade do arco. Esta operação repete-se para o arco de sentido contrário, mas com <LB destino->origem>.

Em seguida, os procedimentos, anteriormente apresentados para um fluxo, são repetidos para os restantes fluxos.

Caso pelo menos um arco do par de percursos determinado tenha carga superior a 100% (passo 14), então é interrompida a ordenação de fluxos (passo 15).



### 3.5.3 – Algoritmo de *Suurballe*

#### 3.5.3.1 – Introdução

Neste subcapítulo vão ser descritas duas implementações do algoritmo de *Suurballe* – uma para determinar pares de percursos disjuntos nas ligações e outra para pares de percursos disjuntos nos nós.

No subcapítulo 2.4, foi referido que no PBB-TE podem ser configurados dois percursos, um de serviço e um de protecção, que necessitam de ser disjuntos para que não haja problemas aquando do *protection switching*. Aqui, o algoritmo de *Suurballe* tem um papel fulcral. Não esquecer que os percursos disjuntos nos nós, são ao mesmo tempo disjuntos nas ligações, por isso, é uma solução mais robusta contra falhas do que a de somente nas ligações.

#### 3.5.3.2 – Algoritmo de *Suurballe* para Par de Percursos Disjuntos nas Ligações

O algoritmo de *Suurballe* para determinar um par de percursos disjuntos nas ligações é dado pelos seguintes passos [18]:

1. Determinar o conjunto  $M1$  de arcos que constitui um percurso de peso mínimo do nó origem  $s$  para o nó destino  $t$ .
2. Determinar os pesos  $c_i$  dos percursos de peso mínimo do nó origem  $s$  para cada um dos outros nós  $i$ .
3. Determinar um novo conjunto de pesos  $w'_{ij}$  para cada arco  $(i,j) \in A$  de acordo com a seguinte transformação:

$$w'_{ij} = \begin{cases} +\infty & , \text{ se } (i,j) \in M_1 \\ w_{ij} + c_i - c_j & , \text{ se } (i,j) \notin M_1 \text{ e } (j,i) \notin M_1 \\ 0 & , \text{ se } (j,i) \in M_1 \end{cases}$$

4. Determinar o conjunto  $M2$  de arcos que constitui um percurso de peso mínimo do nó origem  $s$  para o nó destino  $t$  em que o peso de cada arco é dado pelos pesos  $w'_{ij}$ .
5. Determinar o conjunto  $M = M1 \cup M2$ .
6. Retirar do conjunto  $M$  todos os pares de arcos que representam a mesma ligação, e.g., que tenham os mesmos nós extremos.

Os dois primeiros passos têm como ponto de partida a implementação do algoritmo de *Dijkstra* [19] para determinar um percurso de peso mínimo de um nó origem para todos os nós da rede. Embora o passo 1 seja determinar o percurso de peso mínimo de uma origem para um destino, no passo 2 precisa-se de saber os pesos mínimos dessa origem para os restantes nós. Por esta razão, implementou-se o algoritmo de *Dijkstra* para determinar um percurso de peso mínimo de um nó origem para todos os nós da rede – subcapítulo 3.5.3.4.

Depois de cumpridos os dois primeiros passos do algoritmo de *Suurballe*, procede-se à determinação dos novos pesos segundo a transformação indicada no passo 3 deste algoritmo.

O quarto passo é semelhante ao passo 1, no entanto, o conjunto de arcos a determinar é a partir destes novos pesos calculados. Por isso, repetiu-se o procedimento utilizado no passo 1. Contudo, implementou-se uma nova versão do algoritmo de *Dijkstra* que determina um percurso de peso mínimo de um nó origem para um específico nó destino, isto porque já não é preciso saber as distâncias mínimas entre os vértices e a origem, tornando mais eficiente o desempenho do algoritmo da fase do trabalho que se está a testar.

Finalmente, os dois últimos passos são feitos em conjunto. Para eliminar as ligações comuns aos dois caminhos obtidos implementou-se um algoritmo que começando no segundo arco de um dos caminhos, vai verificar se o arco inverso desse existe no outro caminho. Em caso afirmativo, passa-se a analisar o outro caminho começando pelo arco com origem no destino do arco inverso referido. Em caso contrário, passa ao arco seguinte no primeiro caminho. Este processo é repetido até chegar ao penúltimo nó. O conjunto de nós por onde passa esta análise é um dos percursos disjuntos. O outro percurso obtém-se de igual modo, mas tem-se de começar a análise pelo outro caminho.

A análise é feita a partir do segundo arco até ao penúltimo, pois tanto o primeiro arco como o último, de qualquer dos caminhos, nunca têm o arco inverso correspondente nos caminhos. Isto não ocorre, porque para o vértice origem consideram-se apenas os arcos de saída e para o vértice destino considera-se só arcos de entrada.

Na figura 11, encontra-se esquematizado o método utilizado na determinação dos percursos disjuntos:

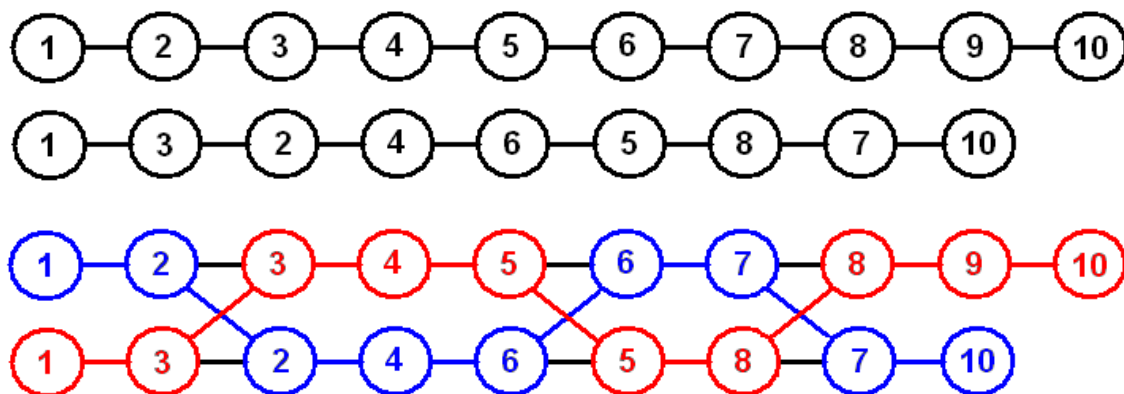


Figura 11. Exemplo de como se remove ligações comuns a dois percursos

As ligações {2,3}, {5,6} e {7,8} eram comuns aos dois caminhos, por isso são eliminadas, resultando em dois percursos disjuntos.

Considerando as seguintes variáveis:

$s$  – vértice origem

$d$  – vértice destino

$h$  – matriz com os dois percursos de peso mínimo

$t_i$  – número de nós pertencentes ao percurso de peso mínimo  $i$

$f$  – matriz com os dois percursos de peso mínimo sem ligações comuns

Este algoritmo que elimina as ligações comuns a dois caminhos pode ser processado pelos seguintes pontos:

```

1   $f_{01} = s$ 
2   $a = 0$ 
3   $b = 1$ 
4   $v = 2$ 
5  for  $i < t_a$  do:
6       $f_{0v} = h_{ai}$ 
7      incrementar  $v$ 
8      for  $j < t_b$  do:
9          if  $h_a(i+1, i) \in h_b$  do:
10             incrementar  $i$  e  $j$ 
11             trocar os valores entre  $i$  e  $j$ 
12              $a = b$ 
13              $b = (b + 1) \% 2$ 
14 if  $h_{ai} \neq d$  do:
15      $f_{0v} = h_{ai}$ 
16     incrementar  $v$ 
17  $f_{0v} = d$ 

```

Como dito anteriormente, este procedimento tem que ser repetido, mas começando pelo outro caminho ( $a = 1$  e  $b = 0$ ) e gravando o percurso na segunda linha da matriz,  $f_{1X}$ .

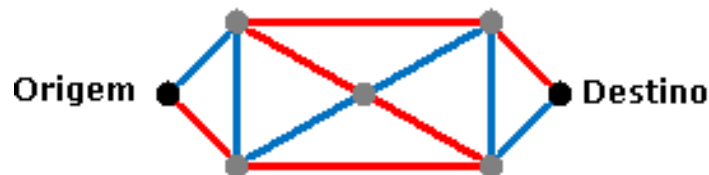


Figura 12. Exemplo de um par de percursos disjuntos nas ligações

### 3.5.3.3 – Algoritmo de *Suurballe* para Par de Percursos Disjuntos nos N  s

O algoritmo de *Suurballe* para determinar um par de percursos disjuntos nos n  s    dado pelos seguintes passos [18]:

1. Determinar o conjunto  $M_1$  de arcos que constitui um percurso de peso m  nimo do n   origem  $s$  para o n   destino  $t$ .
2. Determinar os pesos  $c_i$  dos percursos de peso m  nimo do n   origem  $s$  para cada um dos outros n  s  $i$ .
3. Determinar um novo conjunto de pesos  $w'_{ij}$  para cada arco  $(i,j) \in A$  de acordo com a seguinte transforma  o:

$$w'_{ij} = \begin{cases} +\infty & , \text{ se } (i,j) \in M_1 \\ w_{ij} + c_i - c_j & , \text{ se } (i,j) \notin M_1 \text{ e } (j,i) \notin M_1 \\ 0 & , \text{ se } (j,i) \in M_1 \end{cases}$$

4. Determinar um novo grafo  $G' = (N', A')$  com as seguintes regras:

- i) considerar um segundo n   (designado por n   duplicado) para cada n   interm  dio do percurso definido por  $M_1$  e considerar o conjunto  $N'$  constitu  do por todos os n  s de  $N$  mais todos os n  s duplicados;
- ii) para cada arco  $(i,j)$  do grafo original cujo peso  $w'_{ij} \neq +\infty$ , fazer corresponder no grafo  $G'$  com o mesmo peso  $w'_{ij}$  o arco  $(i, d[i])$  se  $(j,i)$  pertencer a  $M_1$  ou o arco  $(d[i], j)$  se  $(j,i)$  n  o pertencer a  $M_1$  ( $d[i]$  designa o n   duplicado de  $i$  se este n  o tiver sido duplicado ou o pr  prio n    $i$  se este n  o n  o tiver sido duplicado);
- iii) acrescentar ao grafo  $G'$  os arcos  $(d[i], i)$  com peso 0.

5. Determinar o conjunto  $M'_2$  de arcos que constitui um percurso de peso m  nimo do n   origem  $s$  para o n   destino  $t$  no grafo  $G'$ .

6. Determinar  $M_2$  fazendo a correspond  ncia inversa de cada arco de  $M'_2$  no grafo original.

7. Determinar o conjunto  $M = M_1 \cup M_2$ .

8. Retirar do conjunto  $M$  conjunto todos os pares de arcos que representam a mesma liga  o, e.g., que tenham os mesmos n  s extremos.

Analisando estes passos verifica-se que os três primeiros são iguais aos da versão deste algoritmo para as ligações. Como tal, os procedimentos seguidos foram os mesmos.

A estratégia adoptada para fazer o passo 4 foi adicionar os arcos originados pelos nós duplicados no final do *array* que continha essa informação. Implicitamente, aumentou-se o número de vértices de modo a respeitar a alteração provisória na rede.

Em seguida, para esta rede aumentada, executou-se o algoritmo de *Dijkstra*. No passo 6, repõem-se os valores originais dos vértices duplicados.

Finalmente, os dois últimos pontos são executados de igual modo ao caso do algoritmo de *Suurballe* para percursos disjuntos nas ligações.

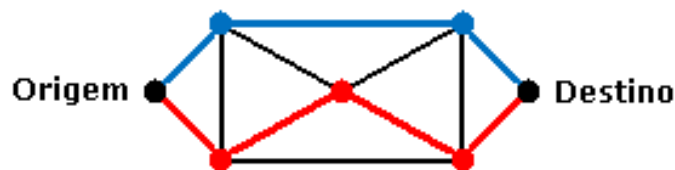


Figura 13. Exemplo de um par de percursos disjuntos nos nós

#### 3.5.3.4 – Algoritmo de *Dijkstra* e construção dos percursos

Um algoritmo eficiente para a determinação de percursos de custo mínimo é o algoritmo de *Dijkstra*. Neste subcapítulo, vai ser apresentado este algoritmo, assim como também a construção de caminhos, baseados nas relações de precedências estabelecidas pelo algoritmo de *Dijkstra*. Os algoritmos descritos neste subcapítulo são utilizados tanto no algoritmo de *Suurballe* para pares de percursos disjuntos nas ligações, como também na versão de pares de percursos disjuntos nos nós [19].

Considerando as seguintes variáveis:

$N$  – número de vértices da rede

$A$  – conjunto de arcos da rede

$s$  – vértice origem

$c_i$  – peso do percurso de peso mínimo do nó origem  $s$  até ao nó  $i$  ( $c_s = 0$ )

$p_i$  – nó predecessor de  $i$  no percurso de peso mínimo do nó origem  $s$  até ao nó  $i$

$S$  – um sub-conjunto dos nós da rede ( $S \subset N$ )

$|S|$  – número de elementos do conjunto  $S$

O algoritmo de *Dijkstra* para determinar um percurso de peso m nimo de um n o origem para todos os destinos de rede pode ser dado pelos seguintes procedimentos [18]:

```

1       $S = \{ \}$ 
2       $c_s = 0$ 
3       $p_s = \text{NULL}$ 
4      for  $i \in N \setminus \{s\}$  do:
5           $c_i = +\infty$ 
6           $p_i = \text{NULL}$ 
7      while  $|S| < N$  do:
8          escolher  $i \notin S$  tal que  $c_i = \min_{j \notin S} c_j$ 
9           $S = S \cup \{i\}$ 
10         for  $(i,j) \in A$  tal que  $j \notin S$  do:
11             if  $c_j > c_i + w_{ij}$  do:
12                  $c_j = c_i + w_{ij}$ 
13                  $p_j = i$ 

```

O ciclo indicado na linha 10 - **for**  $(i,j) \in A$  tal que  $j \notin S$  **do**: - foi implementado recorrendo   Forward Star Form (FSF) [20]. Desta forma, em vez de se percorrer o conjunto de arcos da rede ( $A$ )   procura dos destinos dos arcos com origem em  $i$ , esta busca   apenas efectuada nos arcos com origem  $i$  – subcap tulo 3.5.3.5.

No final da execu  o do algoritmo de *Dijkstra* pode-se aceder aos percursos de peso m nimo dado uma origem. Esta   a origem do fluxo que est  a ser processado neste momento.

Utilizou-se tamb m a vers o do algoritmo de *Dijkstra* que determina um percurso de peso m nimo de um n o origem para um espec fico n o destino. Enquanto na primeira implementa  o percorriam-se os v rtices todos, aqui a fun  o termina quando o v rtice destino acaba de ser analisado. Para esta altera  o surtir efeito alterou-se a linha 7 nos procedimentos do algoritmo de *Dijkstra* para:

**while**  $t \notin S$  **do**:

Como o algoritmo de *Dijkstra* n o devolve caminhos, mas sim estabelece rela  es de preced ncia de peso m nimo, para se saber o percurso de uma origem para um destino   necess rio andar “de tr s para a frente”, ou seja, come ar no destino.

Para isso, implementou-se uma fun  o que parte do v rtice destino, verificando a sua preced ncia, que por sua vez, v  a sua e assim recursivamente at  que o n o predecessor de outro seja o n o origem. A partir daqui guarda-se numa vari vel os v rtices do caminho at  chegar ao n o destino, primeira itera  o.

Considerando as seguintes variáveis:

$s$  – vértice origem

$pi$  – nó predecessor de  $i$  no percurso de peso mínimo do nó origem  $s$  até ao nó  $i$

$rj$  –  $j$ -ésimo vértice do percurso de peso mínimo do nó origem  $s$  até ao nó destino

Esta função pode ser executada nos seguintes passos:

```

1  EscreverCaminho ( $s, i$ ):
2  if  $i = s$  do :
3       $j = 1$ 
4       $rj = i$ 
5      incrementar  $j$ 
6  else
7      EscreverCaminho ( $s, pi$ )
8       $rj = i$ 
9      incrementar  $j$ 

```

Para o caminho 3 – 4 – 6 – 5, o aspecto da recursividade seria a seguinte:

```

EscreverCaminho(3, 5 = destino);

    EscreverCaminho(3, predecessor[5] = 6);

        EscreverCaminho(3, predecessor[6] = 4);

            EscreverCaminho(3, predecessor[4] = 3 = origem);

```

### 3.5.3.5 – Forward Star Form

Como referido atrás, no algoritmo de *Dijkstra* utilizou-se a representação FSF. Esta é uma das mais eficientes estruturas de dados para representar uma rede. Esta representação armazena num único *array* todos os arcos provenientes dos vértices. Na construção desta representação, é atribuído a cada arco, um número de modo a ordenar convenientemente a lista com todos os arcos. A numeração dos arcos começa a ser feita a partir dos que têm origem no vértice 1, depois os com origem no vértice 2 e assim sucessivamente. Uma vez que esta lista de arcos ordenados é obtida, os dados associados aos arcos são armazenados em *arrays*, sequencialmente. Por exemplo, o arco  $(i,j)$ , se tiver o número de arco 3, então os vértices de origem e destino, e a capacidade associados ao arco são armazenados nas seguintes posições dos *arrays*  $origem(3)$ ,  $destino(3)$  e  $capacidade(3)$ .

Para al  m da ordena  o dos arcos, tamb  m    preciso um ponteiro para cada v  rtice  $i$ , representado por  $index(i)$ , que devolve o menor n  mero do arco com origem em  $i$ . No caso de n  o haver arcos com origem no n    $i$ ,  $index(i)$     igualado a  $index(i + 1)$ . Para uma maior coer  ncia desta representa  o, assume-se que  $index(1) = 1$  e  $index(n+1) = m+1$ , sendo que  $n$     o n  mero total de v  rtices da rede e  $m$     o n  mero total de arcos da rede.

Assim, partindo da informa  o das liga  es da rede facilmente preenche-se os *arrays* *origem*, *destino* e *capacidade*, que t  m tamanho igual ao n  mero total de arcos da rede. Depois, com os dados presentes nestes *arrays*, constr  i-se o *array index*.

Considerando as seguintes vari  veis:

$N$  – n  mero de v  rtices da rede

$A$  – n  mero de arcos da rede

$d_i$  – n  mero do primeiro arco com origem no v  rtice  $i$

$b_i$  – v  rtice origem do arco n  mero  $i$

A FSF foi implementada segundo os seguintes procedimentos:

1	$d_1 = 1$
2	$d_{N+1} = A+1$
3	<b>for</b> $i \in N \setminus \{1\}$ <b>do:</b>
4	<b>if</b> $b_j \neq b_{j+1}$ <b>do:</b>
5	$d_i = j+1$
6	incrementar $j$

Em resumo, a FSF    uma estrutura de dados que pode ser utilizada de forma eficaz para determinar o conjunto de arcos de sa  da a partir de qualquer n  , n  o precisando assim de percorrer todos os n  s    procura de um determinado arco.

### 3.5.4 – Melhor Solu  o e Selec  o do Tipo de Percurso

Quando uma ordena  o de fluxos    toda encaminhada, o estado final da rede, isto   , as cargas de todos os arcos,    considerado como uma solu  o v  lida. Nestes casos,    preciso comparar esta solu  o actual com a melhor solu  o encontrada at   ent  o. Para tal, v  o ser comparadas as cargas ordenadas, por ordem decrescente, da solu  o actual com as da melhor solu  o. Se a solu  o actual for melhor que a melhor solu  o, isto   , possuir alguma carga de menor valor que a correspondente na melhor solu  o, sem que antes tenha sido encontrado um caso que esta   ltima fosse melhor que a solu  o actual, ent  o    melhor solu  o vai-se atribuir esta solu  o.



Os dados a guardar relativos à melhor solução são os seguintes:

- Instante de tempo em que a melhor solução foi obtida
- Iteração em que a melhor solução foi obtida
- Cargas da melhor solução
- Cargas ordenadas da melhor solução
- Sequência de todos os pares de percursos obtidos da melhor solução
- Informação dos fluxos da melhor solução
- Expoente da melhor solução

Por último, depois de obtida a melhor solução final, atribui-se para cada par de percursos desta solução qual é o percurso de serviço e o de protecção.

Esta atribuição é feita calculando, para cada caminho no sentido directo e sentido inverso, a sua carga máxima. O que tiver menor carga máxima é escolhido como percurso de serviço e se as cargas máximas forem iguais então é escolhido o caminho com menor número de saltos. Caso estes critérios sejam iguais nos dois percursos, então o primeiro percurso determinado é considerado de serviço e o segundo de protecção.

Considerando as seguintes variáveis:

$NF$  – número de fluxos

$A$  – número de arcos da rede

$max1_i$  – carga máxima do percurso 1 do fluxo  $i$

$max2_i$  – carga máxima do percurso 2 do fluxo  $i$

$t1_i$  – número de saltos do percurso 1 do fluxo  $i$

$t2_i$  – número de saltos do percurso 2 do fluxo  $i$

1	<b>for</b> $i \leq NF$ <b>do:</b>
2	<b>if</b> $max2_i < max1_i$ ou caso $max2_i = max1_i$ , $t2_i < t1_i$ <b>do:</b>
3	Percurso de Serviço $\rightarrow$ Percurso 2
4	Percurso de Protecção $\rightarrow$ Percurso 1
5	<b>else</b>
6	Percurso de Serviço $\rightarrow$ Percurso 1
7	Percurso de Protecção $\rightarrow$ Percurso 2

Denominam-se por percurso de serviço os que, em cada par de percursos disjuntos, encontram-se na primeira posição, enquanto que os de protecção são os segundos. Esta selecção é necessária para a correcta ocorrência do *protection switching*.

## Capítulo 4

### Resultados Computacionais

#### 4.1 – Sumário

Neste capítulo serão apresentados todos os resultados computacionais obtidos. Primeiro, apresentam-se os casos de estudo utilizados, para depois serem detalhados os testes efectuados para cada fase do trabalho, onde esses casos são aplicados.

#### 4.2 – Casos de Estudo

Realizou-se o estudo das redes presentes nas figuras 14, 15 e 16. Estas apresentam a mesma estrutura hierárquica. Assim, num nível inferior, encontram-se os *switches* de acesso, aos quais os equipamentos dos clientes são ligados. Num nível intermédio estão posicionados os *switches* que interligam o nível inferior ao nível superior (considera-se que os equipamentos de clientes empresariais também se ligam a estes *switches*), no topo, está o *edge switch* (*gateway* que liga a rede *Ethernet* a outras redes). As ligações entre os equipamentos dos dois níveis superiores são de 10 Gb/s e as restantes ligações são de 1 Gb/s. Isto justifica-se pelo facto de, no primeiro caso, serem zonas de trânsito, apresentando grandes quantidades de tráfego e, por isso, necessitarem de maior largura de banda.

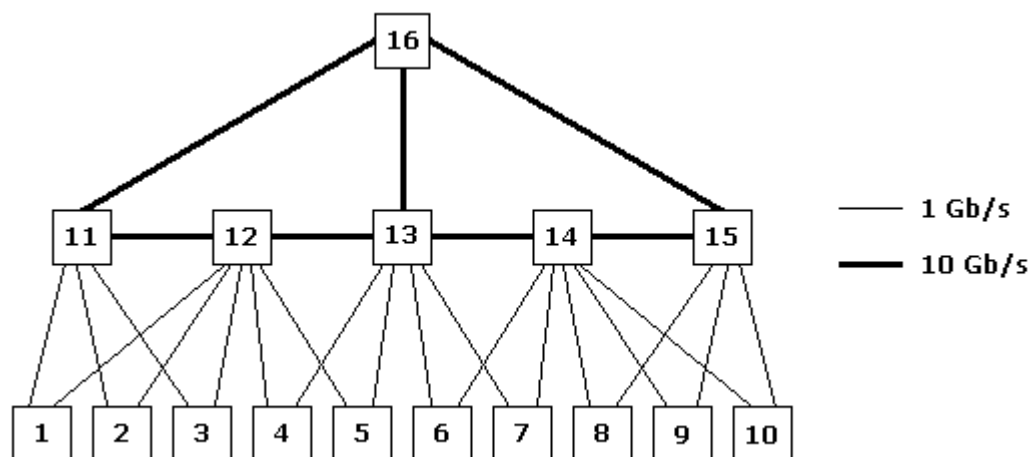


Figura 14. Rede constituída por 16 nós, 27 ligações

Para o estudo da rede da figura 14, consideram-se 90 VLANs e 180 fluxos de tráfego (cada VLAN suporta 2 fluxos de tráfego). Por cada *switch* de acesso para clientes comuns utilizou-se, em média, 6 VLANs para tráfego do *edge switch*, com 2 ou 5 Mb/s de *upstream* e 10 ou 30 Mb/s de *downstream* e duas VLANs para tráfego entre *switches* de clientes comuns, com 20 ou 50 Mb/s, em ambos sentidos. Consideram-se ainda também algumas VLANs dos *switches* intermédios para tráfego entre si, a 100 Mb/s. Esta rede será denominada rede 16.

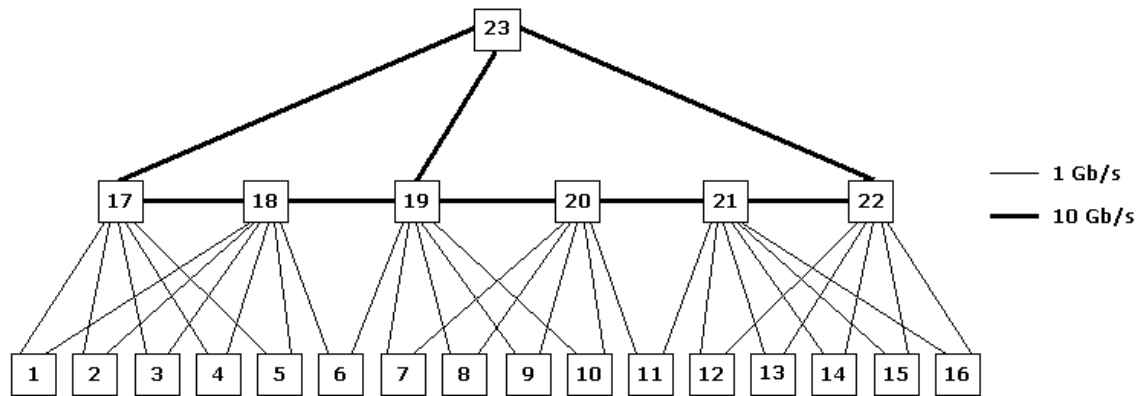


Figura 15. Rede constituída por 23 nós, 40 ligações

Na rede da figura 15, consideram-se 179 VLANs e 358 fluxos de tráfego. Por cada *switch* de acesso para clientes comuns utilizou-se, em média, 8 VLANs para tráfego do *edge switch* e 3 VLANs para tráfego entre *switches* de clientes comuns cujos valores utilizados de *upstream* e *downstream* são os mesmos da rede da figura 14. Também se consideram algumas VLANs dos *switches* intermédios para tráfego entre si, a 100 Mb/s. Esta rede será denominada rede 23.

De modo a ter mais casos de estudo para se obter o maior conjunto de resultados possíveis considerou-se uma rede semelhante à anterior, mas foram acrescentadas mais 24 VLANs, tendo por isso mais 48 fluxos de tráfego. Esta nova rede será denominada rede 23A.

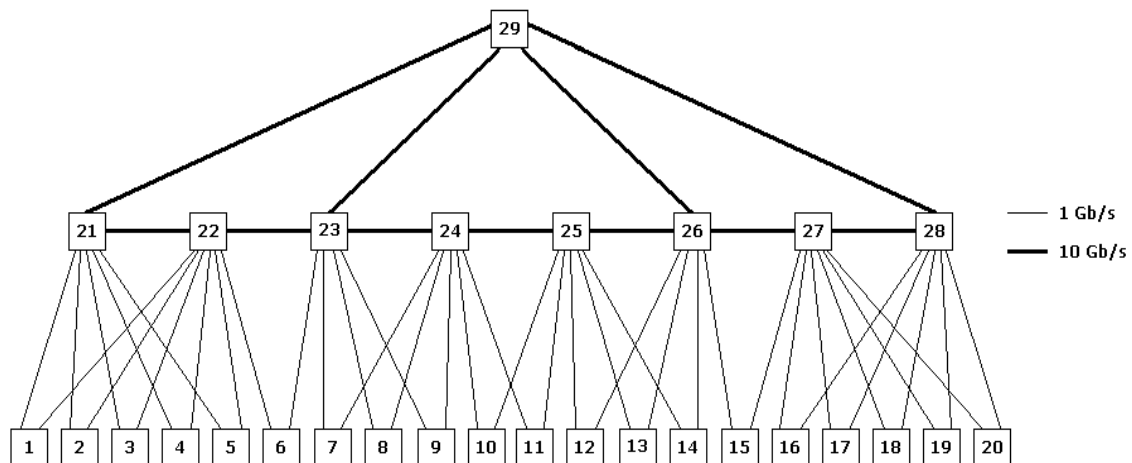


Figura 16. Rede constituída por 29 nós, 51 ligações

Na rede da figura 16, consideram-se 227 VLANs e 454 fluxos de tráfego. O número médio de VLANs utilizado foi semelhante à rede da figura 15. Por cada *switch* de acesso para clientes comuns utilizou-se, em média, 8 VLANs para tráfego do *edge switch* e 3 VLANs para tráfego entre *switches* de clientes comuns, cujos valores utilizados de *upstream* e *downstream* são os mesmos da rede da figura 15. Tal como nos outros casos, também se consideram algumas VLANs dos *switches* intermédios para tráfego entre si, a 100 Mb/s. Esta rede será denominada rede 29.

Pelo mesmo motivo da rede 23A, criou-se uma nova rede 29A com mais 46 VLANs, tendo por isso mais 92 fluxos de tráfego.

Considerou-se assim um total de 10 casos de estudo, pois para cada uma das redes (16, 23, 23A, 29 e 29A) apresentadas consideram-se dois casos: determinação de pares de percursos disjuntos nas ligações e pares de percursos disjuntos nos nós.

Todos os resultados foram obtidos num computador a 1.91 GHz com 512MB de memória RAM.

Antes de executar as várias fases do trabalho, registou-se qual a solução devolvida pelo algoritmo construtivo. É esta solução que vai ser o ponto de partida para as tentativas de melhorar a solução, minimizando as piores cargas, sendo o termo de comparação para com as diversas fases do trabalho.

Estas soluções iniciais, para cada uma das redes apresentadas, podem ser consultadas no Apêndice C (tabelas 25-29).

## 4.3 – Resultados – Fases 1 e 2

Em seguida, realizaram-se um conjunto de testes para verificar em que condições eram obtidas as melhores soluções. Foram consideradas duas opções para cada um dos algoritmos concebidos nas duas primeiras fases do trabalho.

Para a fase 1, Pesquisa Local Aleatória, consideraram-se duas opções: (1) trocar um par de fluxos caso a solução obtida não fosse pior que a melhor solução até então e (2) trocar um par de fluxos caso a solução obtida fosse melhor que a melhor solução obtida até então.

Para a fase 2, Pesquisa por Pesos Modificados, as duas opções criadas consideram dois valores diferentes de *delta* para variação do valor do expoente em cada iteração.

As opções relativas à fase 1 foram executadas 10 vezes cada, sendo que as da fase 2, como as soluções geradas são determinísticas, isto é, os mesmos valores de expoentes resultam nas mesmas soluções, executou-se apenas uma vez cada. Assim, por cada caso de estudo tem-se 22 soluções.

Resumindo foram testadas as seguintes opções:

- Opção 1 – Pesquisa Local Aleatória – Trocar par de fluxos se a solução obtida não piorar a “melhor solução”
- Opção 2 – Pesquisa Local Aleatória – Trocar par de fluxos se a solução obtida melhorar a “melhor solução”
- Opção 3 – Pesquisa por Pesos Modificados –  $\delta = 1$ , 2000 iterações
- Opção 4 – Pesquisa por Pesos Modificados –  $\delta = 0.1$ , 20000 iterações

Não esquecer que nas opções 3 e 4, a gama de valores do expoente é a mesma (1 a 2000), variando apenas o *step* que sofre a cada iteração.

Na primeira vez que se executaram os algoritmos das fases 1 e 2, para além de um número mínimo de iterações, considerou-se também como critério de paragem que sempre que o limite de iterações era atingido, verificava-se se a melhor solução obtida até então tinha sido alcançada na primeira metade das iterações. Em caso afirmativo, considerava-se essa solução como melhor solução final. Caso a melhor solução tivesse sido obtida na segunda metade das iterações, ao número de iterações é adicionado duas unidades e continua-se o método de pesquisa. Este critério de paragem tinha por objectivo executar os algoritmos até ser alcançado o dobro da iteração onde fora obtida a melhor solução, de modo a estar o mesmo número de iterações sem ter melhorado a “melhor solução”.

Contudo este critério revelou-se pouco eficiente pois dependia muito da aleatoriedade das trocas efectuadas na Pesquisa Local Aleatória, sendo que o número total de iterações efectuadas divergia muito, o que dificultava a comparação de resultados, tanto entre soluções da fase 1, como também entre fase 1 e fase 2. Assim, para resolver este problema impôs-se que todas as execuções da fase 1 fizessem o mesmo número de iterações (20000) e retirou-se também esse critério de paragem à fase 2. Depois de alguns testes computacionais, verificou-se que na máquina onde foram obtidos todos os resultados, cada execução dos algoritmos na rede de maior dimensão e de maior número de fluxos de tráfego (rede 29A), 20000 iterações correspondiam a cerca de 5 minutos de tempo total de cálculo, daí a utilização deste valor.

Executaram-se as opções 1, 2, 3 e 4, para os 10 casos de estudo. As tabelas 1-10 apresentam os parâmetros das 22 melhores soluções obtidas para cada caso de estudo. As soluções encontram-se ordenadas, sendo que na primeira linha está a melhor “melhor solução” obtida e na última linha a pior “melhor solução”.

Nestas tabelas estão presentes alguns parâmetros relativos às melhores soluções obtidas:

- Opção utilizada;
- Iteração em que foi obtida a melhor solução;
- Tempo total de execução do algoritmo/fase utilizado;
- Instante em que foi obtida a melhor solução;
- Expoente utilizado quando foi obtida a melhor solução;
- Número de melhores soluções contabilizadas até encontrar a melhor solução final.

As piores cargas das soluções obtidas podem ser consultadas no Apêndice C (tabelas 30-39). Estas cargas encontram-se ordenadas de modo decrescente. Em cada linha encontra-se uma solução, que corresponde aos parâmetros presentes na mesma linha das tabelas 1-10.

Depois de cada uma destas tabelas são apresentadas algumas análises relevantes, retiradas tanto destas, como também das correspondentes tabelas com as piores cargas, apresentadas no Apêndice C. Após as tabelas 1-10 (e respectivas análises parcelares), é feita uma análise mais global aos resultados obtidos.

Opção	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
1	2740	37	4	1	25
1	1966	35	3	1	29
1	1563	33	2	1	31
1	3363	32	5	1	31
1	4092	32	6	1	24
1	8222	34	13	1	27
1	1693	37	3	1	25
1	2169	35	3	1	27
2	2079	36	3	1	18
3	291	7	0	292	3
4	2909	72	7	291,9	3
1	18379	34	32	1	30
1	19994	35	35	1	16
2	1144	37	1	1	5
2	437	60	0	1	5
2	127	38	0	1	3
2	216	36	0	1	3
2	261	37	0	1	3
2	373	36	0	1	3
2	25	35	0	1	2
2	1152	31	1	1	3
2	90	34	0	1	3

Tabela 1. Resultados relativos à rede 16, para pares de percursos disjuntos nas ligações – Fases 1 e 2

Opção	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
1	2042	37	3	1	23
1	1289	37	2	1	21
1	711	38	1	1	19
1	1564	38	2	1	20
1	1684	37	3	1	20
1	12623	37	23	1	22
1	1355	37	2	1	26
1	720	40	1	1	20
1	1943	38	3	1	22
1	999	40	2	1	27
3	284	8	0	285	2
4	2835	77	7	284,5	7
2	1083	42	2	1	4
2	4	41	0	1	1
2	1	43	0	1	1
2	9	40	0	1	1
2	24	42	0	1	1
2	7	40	0	1	1
2	5	38	0	1	1
2	1	40	0	1	1
2	2	38	0	1	1
2	41	44	0	1	2

Tabela 2. Resultados relativos à rede 16, para pares de percursos disjuntos nos nós – Fases 1 e 2

Análises relevantes das tabelas 1 e 2:

- As cargas de todas as soluções são muito semelhantes (tabelas 30 e 31 – Apêndice C);
- A Opção 1 é melhor que a Opção 2;
- As melhores soluções da Opção 3 e da Opção 4 são iguais;
- A Opção 4 demora mais tempo que Opção 3 a encontrar a melhor solução, porque tem de executar muitos expoentes até obtê-la;
- As Opções 3 e 4 têm expoentes muito elevados.

Opção	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
1	18970	105	100	1	114
1	18838	108	102	1	146
1	19410	102	99	1	155
1	15540	103	80	1	140
1	17144	102	88	1	154
1	17261	108	94	1	121
1	19271	110	106	1	102
1	18593	103	95	1	111
1	18945	105	100	1	159
1	19127	103	98	1	127
2	17265	113	92	1	103
2	11123	105	58	1	78
2	15331	106	82	1	75
2	18550	103	95	1	89
2	2024	102	10	1	71
2	18660	103	95	1	77
2	15167	111	85	1	64
2	12826	107	70	1	59
2	17453	101	88	1	60
2	7289	111	40	1	69
4	4	226	0	1,4	4
3	1	22	0	2	1

Tabela 3. Resultados relativos à rede 23, para pares de percursos disjuntos nas ligações – Fases 1 e 2

Análises relevantes:

- A Opção 1 é claramente melhor que a Opção 2;
- Na Pesquisa por Pesos Modificados, a sua pior carga é 2,1% pior que a melhor solução da Pesquisa Local Aleatória (tabela 32 – Apêndice C);
- A Pesquisa por Pesos Modificados demora menos de 1 segundo a atingir a melhor solução;
- A Opção 4 é melhor que a Opção 3;
- As opções 4 e 3 têm os expoentes 1,4 e 2, respectivamente.



Opção	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
1	18585	117	109	1	138
1	19442	122	118	1	129
1	18559	118	109	1	144
1	19505	118	115	1	117
1	19206	117	113	1	104
1	17981	117	105	1	136
1	19795	119	118	1	136
1	19407	122	118	1	168
1	16238	116	94	1	129
1	9724	115	55	1	128
2	10772	123	65	1	69
2	17236	115	100	1	71
2	8082	121	48	1	84
2	10743	116	62	1	78
2	5601	124	33	1	87
2	16188	124	102	1	76
2	8758	116	51	1	78
2	1364	121	7	1	61
4	6	228	0	1,6	6
3	431	22	3	432	12
2	16048	119	95	1	29
2	1696	115	9	1	19

Tabela 4. Resultados relativos à rede 23, para pares de percursos disjuntos nos nós – Fases 1 e 2

Análises relevantes:

- A Opção 1 é claramente melhor que a Opção 2;
- Na Pesquisa por Pesos Modificados, a sua pior carga é 1,1% pior que a melhor solução da Pesquisa Local Aleatória (tabela 33 – Apêndice C);
- A Opção 4 demora menos de 1 segundo a atingir a melhor solução;
- A Opção 4 é melhor que a Opção 3;
- As opções 4 e 3 têm os expoentes 1,6 e 432, respectivamente.

Opção	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
1	17987	148	122	1	126
1	18539	133	124	1	136
1	19694	119	117	1	170
1	14885	172	128	1	149
1	13381	114	76	1	122
1	15213	128	97	1	142
1	19331	165	160	1	151
1	19244	131	125	1	134
1	19946	126	126	1	131
1	16370	172	151	1	156
2	14697	114	83	1	93
2	15907	113	90	1	124
2	18494	113	105	1	116
2	14804	117	87	1	108
2	19681	115	113	1	94
2	13719	125	84	1	65
2	3468	115	19	1	70
2	14466	114	82	1	80
2	19920	117	117	1	123
2	6112	120	35	1	42
4	4	239	0	1,4	4
3	287	23	2	288	3

Tabela 5. Resultados relativos à rede 23A, para pares de percursos disjuntos nas ligações – Fases 1 e 2

## Análises relevantes:

- A Opção 1 é claramente melhor que a Opção 2;
- Na Pesquisa por Pesos Modificados, a sua pior carga é 3,5% pior que a melhor solução da Pesquisa Local Aleatória (tabela 34 – Apêndice C);
- A Opção 4 demora menos de 1 segundo a atingir a melhor solução;
- A Opção 4 é melhor que a Opção 3;
- As opções 4 e 3 têm os expoentes 1,4 e 288, respectivamente.

Opção	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
1	19700	148	146	1	174
1	17874	131	117	1	159
1	19112	136	130	1	201
1	19034	138	128	1	175
1	19609	171	167	1	187
1	19975	138	138	1	140
1	17527	140	123	1	175
1	17417	148	127	1	130
1	15936	142	113	1	139
1	17925	167	152	1	145
2	7778	129	50	1	87
4	7	341	0	1,7	6
3	1	33	0	2	1
2	482	130	3	1	19
2	4029	128	25	1	28
2	4522	128	29	1	20
2	8747	128	56	1	23
2	19739	129	127	1	21
2	586	128	3	1	13
2	764	128	4	1	17
2	7781	129	51	1	13
2	1048	128	6	1	16

Tabela 6. Resultados relativos à rede 23A, para pares de percursos disjuntos nos nós – Fases 1 e 2

## Análises relevantes:

- A Opção 1 é claramente melhor que a Opção 2;
- Na Pesquisa por Pesos Modificados, a sua pior carga é 1% pior que a melhor solução da Pesquisa Local Aleatória (tabela 35 – Apêndice C);
- A Pesquisa por Pesos Modificados demora menos de 1 segundo a atingir a melhor solução;
- A Opção 4 é melhor que a Opção 3;
- As opções 4 e 3 têm os expoentes 1,7 e 2, respectivamente.

Opção	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
1	19697	165	162	1	142
1	19676	165	162	1	131
1	19699	165	162	1	139
1	18769	165	155	1	133
1	18608	165	153	1	149
1	19009	165	157	1	105
1	19170	165	158	1	156
1	19051	165	157	1	147
1	19894	165	164	1	141
1	15659	165	129	1	111
2	13510	165	111	1	102
2	19720	165	163	1	129
2	15134	165	125	1	115
2	18874	165	155	1	105
2	5294	165	43	1	96
2	18372	165	151	1	82
2	17347	164	142	1	102
2	19440	165	160	1	101
2	12737	165	105	1	114
2	2127	165	17	1	66
4	4	321	0	1,4	4
3	1	32	0	2	1

Tabela 7. Resultados relativos à rede 29, para pares de percursos disjuntos nas ligações – Fases 1 e 2

Análises relevantes:

- A Opção 1 é claramente melhor que a Opção 2;
- Na Pesquisa por Pesos Modificados, as oito piores cargas são iguais à melhor solução da Pesquisa Local Aleatória (tabela 36 – Apêndice C);
- A Pesquisa por Pesos Modificados demora menos de 1 segundo a atingir a melhor solução;
- A Opção 4 é melhor que a Opção 3;
- As opções 4 e 3 têm os expoentes 1,4 e 2, respectivamente.

Opção	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
3	520	36	8	521	19
4	5192	369	82	520,2	34
1	19212	186	178	1	179
1	19655	186	183	1	209
1	17474	185	162	1	166
1	19907	185	185	1	198
1	19253	185	178	1	193
1	17732	185	164	1	166
1	18136	186	168	1	170
1	16046	186	149	1	178
1	16215	185	150	1	169
1	17211	186	160	1	141
2	19406	185	179	1	135
2	13846	185	128	1	101
2	18361	184	169	1	118
2	3912	184	36	1	90
2	8594	184	79	1	87
2	6483	185	60	1	80
2	18645	185	172	1	95
2	10114	184	93	1	98
2	19664	185	182	1	95
2	15231	185	141	1	67

Tabela 8. Resultados relativos à rede 29, para pares de percursos disjuntos nos nós – Fases 1 e 2

Análises relevantes:

- A Opção 1 é claramente melhor que a Opção 2;
- A Pesquisa por Pesos Modificados é melhor que a Pesquisa Local Aleatória;
- As melhores soluções da Opção 3 e da Opção 4 são iguais;
- A Opção 4 demora mais tempo que Opção 3 a encontrar a melhor solução, porque tem de executar muitos expoentes até obtê-la;
- As Opções 3 e 4 têm expoentes muito elevados.

Opção	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
1	19250	199	192	1	164
1	19901	200	199	1	147
1	18734	199	187	1	141
1	19964	199	199	1	163
1	18495	199	184	1	162
1	19620	199	195	1	143
1	18201	199	181	1	157
1	19781	199	197	1	180
1	17948	199	179	1	132
1	19635	199	195	1	185
2	19344	199	193	1	104
2	19519	198	194	1	130
2	15899	199	158	1	82
2	13196	199	131	1	92
2	16539	198	164	1	79
2	15776	199	157	1	107
2	15113	199	150	1	89
2	17254	199	171	1	91
2	2936	199	29	1	76
2	3305	197	32	1	63
3	0	34	0	1	0
4	0	343	0	1	0

Tabela 9. Resultados relativos à rede 29A, para pares de percursos disjuntos nas ligações – Fases 1 e 2

Análises relevantes:

- A Opção 1 é claramente melhor que a Opção 2;
- A Pesquisa Local Aleatória é melhor que a Pesquisa por Pesos Modificados;
- Na Pesquisa por Pesos Modificados, as duas piores cargas são 5% piores que as da melhor solução da Pesquisa Local Aleatória (tabela 38 – Apêndice C);
- A Pesquisa por Pesos Modificados nunca melhora a solução inicial do algoritmo construtivo.

Opção	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
3	618	44	12	619	22
4	6174	447	120	618,4	29
1	18677	224	209	1	217
1	17910	224	200	1	213
1	16937	224	189	1	235
1	19761	224	222	1	287
1	19254	225	216	1	224
1	19641	224	220	1	219
1	19488	224	218	1	263
1	19955	225	224	1	169
1	15093	224	169	1	180
2	19379	223	216	1	146
1	18748	223	209	1	211
2	17356	223	193	1	111
2	19437	224	218	1	130
2	16413	224	184	1	90
2	19814	223	221	1	84
2	8849	223	98	1	56
2	8182	223	91	1	38
2	19112	224	214	1	63
2	17463	224	195	1	52
2	2970	224	33	1	41

Tabela 10. Resultados relativos à rede 29A, para pares de percursos disjuntos nos nós – Fases 1 e 2

## Análises relevantes:

- A Opção 1 é melhor que a Opção 2;
- A Pesquisa por Pesos Modificados é melhor que a Pesquisa Local Aleatória;
- As melhores soluções da Opção 3 e da Opção 4 são iguais;
- A Opção 4 demora mais tempo que Opção 3 a encontrar a melhor solução, porque tem de executar muitos expoentes até obtê-la;
- As Opções 3 e 4 têm expoentes muito elevados.

Antes de começar a analisar as tabelas 1-10, vale a pena referir que praticamente todos os testes realizados (excepto na rede 29A, para percursos disjuntos nas ligações, nas opções 3 e 4), melhoraram a respectiva solução inicial obtida pelo algoritmo construtivo. Se para a rede 16, as melhorias não foram significativas, para as redes maiores conseguiram-se melhorias entre 5 e 6,7% logo na pior carga.

Passando à análise das tabelas 1-10, a primeira conclusão que se pode retirar é que 98% das soluções originadas pela Opção 1 são melhores que a “melhor solução” originada pela Opção 2. Isto ocorre pois a Opção 2 limita o número de trocas de fluxos possível, uma vez que, apenas troca quando a solução é melhor e, em muitos casos, observou-se que as soluções originadas eram iguais à melhor solução encontrada até então. De referir ainda que a Opção 2 nunca resulta em melhores soluções do que a melhor da Opção 1. Na Pesquisa Local Aleatória, verifica-se a tendência de que quantas mais vezes a “melhor solução” for melhorada, melhor é essa solução em relação a outras execuções.

Comparando as Opções 3 e 4, verifica-se que quando o valor do expoente da melhor solução é muito elevado ( $> 100$ ) para ambos os valores de *delta*, o valor deste não tem muita influência, visto que as soluções originadas são iguais (tabelas 1, 2, 8 e 10). Contudo, a Opção 4 como tem um *delta* inferior demora mais, pois executa demasiados expoentes até encontrar a melhor solução. Quando o valor do expoente da melhor solução da Opção 4 é baixo e diferente do da Opção 3, a Opção 4 é sempre melhor (tabelas 3-7). Na tabela 9, verifica-se que a solução inicial nunca é melhorada, registando o melhor expoente valor 1.

A partir daqui só serão considerados 8 casos de estudo, visto que as soluções da rede 16 serem muito idênticas e não proporcionarem resultados relevantes de serem analisados, pois a solução inicial já é muito boa.

Na tabela 11, encontram-se resumidos os principais resultados comparativos entre a Opção 1 e a Opção 4. Faz-se a comparação destas opções visto serem os métodos que geram melhores soluções em cada uma das fases em estudo (Opção 1 é claramente melhor que a Opção 2 e a Opção 4 gera soluções iguais ou melhores que a Opção 3).

Nº de casos de estudo em que ...	Método	Opção 1	Opção 4
... gera a melhor solução total		6	2
... há diferenças entre as piores cargas dos dois métodos		5	—
... obtém melhor desempenho		5	3

Tabela 11. Opção 1 Vs Opção 4



Ao comparar os resultados apresentados na tabela 11, comparando a Opção 1 com a Opção 4, verifica-se que apenas em dois casos de estudo é que esta última pesquisa originou a melhor solução de todas (tabelas 8 e 10) - melhor solução total.

No entanto, dos 6 casos de estudo em que a Opção 1 gera melhores soluções totais, existem 5 que apresentam diferenças nas piores cargas em relação à Opção 4. Nestes casos, em que o valor do expoente da melhor solução é pequeno, apesar de se constatar que as soluções originadas pela Opção 4 são, no máximo, 5% piores nas cargas mais elevadas do que as soluções da Opção 1, são obtidas em muito menos iterações, no máximo, em apenas 7, que corresponde a menos de 1 segundo. Nestes 6 casos de estudo, há 1 em que não existem diferenças nas piores cargas das duas opções. Contudo, neste caso, a Opção 4 tem um desempenho melhor, porque obtém a solução num tempo mais reduzido que na Opção 1.

Embora as soluções obtidas através da Pesquisa por Pesos Modificados sejam conseguidas em menos iterações, no que diz respeito ao tempo total, isso não acontece, para o mesmo número de iterações. O tempo que demora a executar uma iteração da Pesquisa por Pesos Modificados é sensivelmente o dobro da que demora uma iteração na Pesquisa Local Aleatória. Isto acontece porque a cada iteração da fase 2, antes de se calcular o par de percursos disjuntos para um dado fluxo, é sempre necessário aplicar um expoente a todos os arcos da rede, aumentando assim a carga computacional do programa.

Assim, para os casos de estudo considerados, os resultados obtidos anteriormente não demonstram que um método é claramente melhor que o outro, mas pode-se concluir que a Pesquisa Local Aleatória origina melhores soluções, enquanto que a Pesquisa por Pesos Modificados obtém soluções próximas, em menos iterações e, consequentemente, em tempos de execução muito reduzidos.

Como tal, na fase 3 deste trabalho procura-se conjugar estes dois métodos de pesquisa tentando ainda melhorar estas “melhores soluções” obtidas ou encontrar soluções próximas, reduzindo o número de iterações, em relação às fases anteriores. Contudo, pode-se afirmar que, na Pesquisa Local Aleatória, a Opção 1 é claramente melhor que a Opção 2. No que diz respeito à Pesquisa por Pesos Modificados, a Opção 4 gera soluções iguais ou melhores que a Opção 3, no entanto, para valores de expoente elevados, esta última opção obtém melhor comportamento, pois executa menos iterações, por ter um *delta* maior.

## 4.4 – Resultados – Fase 3

Para testar a fase 3 e de acordo com os resultados obtidos, efectivou-se a Opção 1 no algoritmo desta fase e alterou-se a Pesquisa por Pesos Modificados. No algoritmo deste tipo de pesquisa, os expoentes a aplicar aos pesos dos arcos eram aplicados de forma linear, por terem um *delta* fixo ao longo da execução do algoritmo. Mediante os resultados nas fases 1 e 2, verifica-se que para expoente pequenos, com um *delta* de 0.1 foram obtidos melhores resultados do que com um *delta* maior. Para expoentes elevados verificou-se que o valor de *delta* não tinha grande influência. De acordo com estas conclusões, criaram-se as seguintes opções, variando os valores de *delta* de acordo com:

➤ Exp 5

Gama de expoentes	<i>delta</i>
[1 ; 5]	0.1

➤ Exp 10

Gama de expoentes	<i>delta</i>
[1 ; 5]	0.1
]5 ; 10]	0.5

➤ Exp 1000

Gama de expoentes	<i>delta</i>
[1 ; 5]	0.1
]5 ; 10]	0.5
]10 ; 1000]	1

Com estas opções, os expoentes a aplicar aos pesos dos arcos passam a ser variáveis por deixarem de ter um *delta* fixo e passarem a ter um *delta* variável. Utilizou-se:

- uma opção para expoentes pequenos (Exp 5);
- uma opção intermédia aumentando ligeiramente a gama de expoentes anterior, adicionando uma nova gama de valores afectados por um *delta* entre 0.1 e 1 (Exp 10);
- uma opção com uma grande gama de expoentes, acrescentando a Exp 10 uma nova gama com *delta* 1, que contempla os valores mais elevados de expoentes (Exp 1000).

Em seguida, executaram-se 10 vezes cada uma das opções apresentadas (Exp 5, Exp 10 e Exp 1000), para os 8 casos de estudo, com um limite de 20000 iterações – tabelas 12-19.

As piores cargas das soluções podem ser consultadas no Apêndice C (tabelas 40-47). Os resultados encontram-se organizados nos mesmos formatos utilizados anteriormente.

De igual modo às fases 1 e 2, depois de cada uma das tabelas 12-19 são apresentadas algumas análises relevantes, retiradas tanto destas, como também das correspondentes tabelas com as piores cargas, apresentadas no Apêndice C. Após estas tabelas (e respectivas análises), é feita uma análise mais global aos resultados obtidos.

Exp	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
10	13211	148	98	8	21
5	11868	148	88	5	17
5	11914	148	88	4,3	18
10	18295	147	134	7	33
5	14322	148	106	4	20
5	19219	147	142	5	35
5	10851	147	80	4,4	18
10	19840	148	147	3,9	20
5	17417	147	128	3	35
10	16272	148	120	4,6	18
10	17986	148	133	2,8	8
10	11874	148	88	4,7	14
5	8749	148	65	4,8	23
5	4256	148	31	3,4	8
10	19608	148	145	4,7	13
10	8824	148	65	2,6	13
5	18019	148	133	2,4	26
10	7932	148	58	3,4	7
10	14453	147	106	2,2	15
5	9356	148	69	2	10
1000	18762	196	184	2,3	6
1000	17729	195	173	1,9	6
1000	12515	195	122	1,9	4
1000	18749	195	183	1,9	6
1000	3136	195	30	1,9	2
1000	4172	196	40	1,6	1
1000	1047	196	10	1,6	1
1000	13547	196	132	1,6	3
1000	17711	195	173	1,6	3
1000	10419	195	102	1,6	3

Tabela 12. Resultados relativos à rede 23, para pares de percursos disjuntos nas ligações – Fase 3

Análises relevantes:

- Exp 5 e Exp 10 apresentam as melhores soluções;
- Exp 5 e Exp 10 obtêm resultados semelhantes, não havendo um que seja claramente melhor que o outro;
- Exp 1000 não tem nenhuma solução entre as 20 melhores soluções.

Exp	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
5	14507	163	119	3,9	32
5	16879	165	139	3,7	31
1000	19205	217	208	423	5
10	17351	164	143	5,5	9
5	18662	164	153	4,3	22
5	18276	165	151	4,8	28
10	10310	163	84	6	19
10	14386	165	119	5,5	15
5	8998	165	74	5	17
5	12017	165	99	4,7	14
5	17881	163	146	3,2	22
5	18429	164	151	4,8	25
10	12909	163	105	4,9	16
10	15452	163	126	3,7	11
10	11824	165	97	5,5	21
10	7013	165	58	4,3	33
10	8310	164	68	3,8	10
5	9145	165	75	3,5	18
10	16148	163	132	3,6	17
10	14866	164	122	2,5	24
5	6532	166	54	2,6	16
1000	17715	217	192	2,1	7
1000	17711	217	192	2,2	4
1000	17718	217	193	2,3	3
1000	6258	216	67	2,2	4
1000	3131	217	34	1,8	1
1000	19001	217	206	219	3
1000	5213	216	56	1,7	1
1000	19791	217	215	1,7	2
1000	13553	216	147	1,6	2

Tabela 13. Resultados relativos à rede 23, para pares de percursos disjuntos nos nós – Fase 3

Análises relevantes:

- Exp 5 e Exp 10 apresentam as melhores soluções;
- Exp 5 é predominante nas 10 melhores soluções;
- Exp 1000 tem apenas uma solução entre as 20 melhores soluções.

Exp	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
5	16539	171	141	4,9	17
5	5820	170	49	4,9	23
10	11368	171	97	5	23
5	7504	170	64	4,5	14
10	14506	171	124	5,5	17
10	7434	174	65	10	30
5	17314	169	146	3,5	36
5	19993	171	171	4,5	54
10	15161	170	129	3,5	31
5	16303	171	140	3,2	29
5	17895	170	153	2,6	29
10	14359	175	126	3	22
5	14843	170	126	3,8	34
10	10688	171	91	4,2	27
10	9294	170	79	6,5	18
5	19449	170	165	3,8	24
10	19050	173	165	2,6	20
10	15460	171	133	3,6	17
10	11551	172	100	3,1	26
1000	9382	227	107	2,2	2
5	17993	170	153	2,2	15
1000	16673	220	184	2,5	9
1000	18759	221	207	2,2	7
1000	19793	220	218	1,9	3
1000	14594	221	161	2,4	8
1000	19801	219	218	2,8	2
1000	16673	225	189	2,1	4
1000	19796	222	220	2,1	4
1000	15624	221	172	1,8	6
1000	17708	221	196	1,9	7

Tabela 14. Resultados relativos à rede 23A, para pares de percursos disjuntos nas ligações – Fase 3

## Análises relevantes:

- Exp 5 e Exp 10 apresentam as melhores soluções;
- Exp 5 e Exp 10 obtêm resultados semelhantes, não havendo um que seja claramente melhor que o outro;
- Exp 1000 tem apenas uma solução entre as 20 melhores soluções.

Exp	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
10	10804	187	101	4,9	31
1000	9922	245	120	509	3
10	18165	186	169	6	25
10	19864	187	186	7	29
10	6499	187	60	6,5	26
5	5150	188	48	5	26
5	15776	187	147	5	27
5	18836	189	179	4,7	37
5	7014	189	66	5	19
10	5937	187	55	5,5	23
5	19001	192	182	4,4	17
10	19682	187	184	6,5	34
5	17558	191	168	5	31
5	19368	187	181	5	23
5	7024	190	66	5	36
10	12177	187	114	4,5	24
5	12444	190	118	4,8	22
10	14953	187	140	4,1	33
1000	14007	243	170	430	4
10	16026	187	150	5,5	26
10	13270	187	123	8,5	34
5	16011	188	150	4	15
1000	16672	244	204	2,2	3
1000	18753	244	229	2,2	6
1000	18039	244	220	291	3
1000	16680	244	203	2,8	5
1000	17713	243	216	2,3	1
1000	17713	244	216	1,9	2
1000	19798	243	242	1,9	3
1000	10428	244	127	2,1	3

Tabela 15. Resultados relativos à rede 23A, para pares de percursos disjuntos nos nós – Fase 3

Análises relevantes:

- Exp 5 e Exp 10 apresentam as melhores soluções;
- Exp 5 e Exp 10 obtêm resultados semelhantes, não havendo um que seja claramente melhor que o outro;
- Exp 1000 tem apenas duas soluções entre as 20 melhores soluções.

Exp	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
5	18017	248	223	2,1	34
10	15446	248	192	2	42
10	13522	246	167	2	29
5	15435	247	191	2,1	25
5	16721	248	207	1,7	37
5	19098	248	237	1,2	49
5	19915	247	246	1,1	40
10	16987	248	211	2,4	37
10	18387	248	228	1,4	28
5	19667	247	243	1,1	41
10	13237	246	163	2	38
5	19908	250	249	1,3	39
10	19717	246	242	1	29
10	14869	247	184	2,1	26
10	9980	248	124	2	20
5	19633	248	244	1,2	46
5	11019	247	136	1,9	31
10	9411	248	117	1,8	16
5	19924	247	246	1,1	37
10	18484	250	231	1,1	24
1000	19792	325	322	1,7	4
1000	17716	323	287	1,5	6
1000	17712	324	288	1,4	2
1000	15626	326	255	1,6	4
1000	19788	326	323	1,6	5
1000	18750	323	304	1,6	6
1000	17706	325	288	1,5	5
1000	10423	325	169	1,6	3
1000	19796	325	323	2,1	7
1000	15632	325	254	1,4	5

Tabela 16. Resultados relativos à rede 29, para pares de percursos disjuntos nas ligações – Fase 3

## Análises relevantes:

- Exp 5 e Exp 10 apresentam as melhores soluções;
- Exp 5 e Exp 10 obtêm resultados semelhantes, não havendo um que seja claramente melhor que o outro;
- Exp 1000 não tem nenhuma solução entre as 20 melhores soluções.



Exp	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
1000	17204	356	306	507	9
1000	12006	355	212	511	6
1000	14086	355	249	511	7
1000	16170	355	287	514	4
1000	6805	356	120	516	4
1000	13051	355	231	517	4
1000	16174	354	286	519	4
1000	9937	355	175	519	4
1000	1602	355	27	521	1
1000	1602	355	27	521	1
5	16762	271	227	2,6	33
5	17998	270	243	2,5	26
10	15922	273	218	3,7	21
10	19614	272	267	2,1	21
5	18062	271	245	2,5	24
10	9937	271	135	3,2	17
5	17598	270	238	1,8	22
5	14010	272	191	1,6	18
10	11062	271	150	2	12
10	19905	272	270	1,5	12
10	19230	271	260	1,3	21
5	12473	272	170	2,3	14
5	18249	270	246	1,7	30
5	7693	269	104	2,2	15
5	19418	269	261	1,3	19
10	19865	272	270	1,9	23
10	18772	270	254	1,4	23
10	9769	271	133	1,6	13
10	18695	275	258	1,4	24
5	11249	272	153	1,5	16

Tabela 17. Resultados relativos à rede 29, para pares de percursos disjuntos nos nós – Fase 3

Análises relevantes:

- Exp 1000 é claramente melhor;
- Exp 5 e Exp 10 obtêm resultados semelhantes, não havendo um que seja claramente melhor que o outro;
- Exp 5 e Exp 10 não têm nenhuma solução nas 10 melhores soluções.

Exp	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
5	18234	299	273	4,3	66
5	19370	299	289	1,7	38
10	19254	299	288	2,5	55
10	12406	300	187	2,5	43
5	19977	298	298	2,2	65
5	19036	298	283	1,6	54
10	16864	299	252	2,3	49
5	16471	299	246	2,1	56
5	16808	299	251	1,5	48
5	17838	298	266	2,1	33
10	17263	299	259	3,4	29
10	19969	300	300	1,2	50
10	16798	299	251	1,7	35
5	19564	298	292	1,3	46
5	18974	299	284	1,5	46
10	17340	299	259	1,2	55
10	17781	300	266	1,1	20
5	15442	298	230	1,2	34
10	18369	299	274	1,2	41
10	19681	299	294	1	48
1000	16661	381	318	1	7
1000	14579	379	277	1,1	4
1000	19789	380	377	1	4
1000	12498	380	238	1	3
1000	18752	381	358	1	5
1000	7287	379	138	1	4
1000	13542	379	257	1,1	4
1000	13538	380	257	1	6
1000	10419	379	198	1,4	3
1000	18742	379	356	1,1	5

Tabela 18. Resultados relativos à rede 29A, para pares de percursos disjuntos nas ligações – Fase 3

Análises relevantes:

- Exp 5 e Exp 10 apresentam as melhores soluções;
- Exp 5 é predominante nas 10 melhores soluções;
- Exp 1000 não tem nenhuma solução entre as 20 melhores soluções.

Exp	Iteração da Melhor Solução	Tempo Total (s)	Instante da Melhor Solução (s)	Expoente da Melhor Solução	Nº Melhorias da Melhor Solução
1000	8990	422	188	621	3
1000	10028	421	210	617	4
1000	3782	421	78	619	2
1000	7939	421	166	611	3
1000	6905	421	144	617	2
1000	11061	421	232	607	2
1000	19390	421	408	610	7
1000	3775	421	78	612	3
1000	13155	422	277	621	3
1000	18352	421	386	615	1
10	11156	327	183	8,5	30
10	15962	327	261	5	27
5	8849	325	144	5	31
5	19962	323	322	1,7	37
10	14428	325	235	1,9	26
5	17838	324	289	3,3	41
10	19159	325	312	3	32
5	14561	324	236	3,4	39
10	19861	325	323	2,7	40
10	14646	325	238	2,6	18
5	19974	325	324	1,7	30
10	19350	324	313	2,4	47
10	10205	324	166	1,2	22
5	19369	326	316	2	31
10	18970	326	309	2,3	23
5	16092	328	264	1,7	22
10	18030	327	295	6	34
5	16502	324	268	2,4	22
5	19055	325	309	1,4	17
5	17214	325	280	1,6	25

Tabela 19. Resultados relativos à rede 29A, para pares de percursos disjuntos nos nós – Fase 3

## Análises relevantes:

- Exp 1000 é claramente melhor;
- Exp 5 e Exp 10 obtêm resultados semelhantes, não havendo um que seja claramente melhor que o outro;
- Exp 5 e Exp 10 não têm nenhuma solução nas 10 melhores soluções.

Nas tabelas 12-19, a coluna respeitante ao número de melhorias da melhor solução tem um significado diferente em relação aos resultados das fases 1 e 2. Enquanto nestas fases sempre que se encontrava uma melhor solução esta alteração era contabilizada, na fase 3 as melhores soluções só são contabilizadas depois de terminar o par de pesquisas, isto é, Pesquisa Local Aleatória seguida de Pesquisa por Pesos Modificados.

Nas tabelas 20 e 21, os casos de estudo em que foram determinados pares de percursos disjuntos nas ligações serão identificados por DL, já os casos em que foram determinados pares de percursos disjuntos nos nós serão identificados por DN.

A tabela 20 apresenta as melhorias, de cada caso de estudo, em relação às melhores soluções das fases 1 e 2, e como em dois casos de estudo (rede 29 – DL e rede 29A – DL) estas soluções nunca são melhoradas, retiraram-se os valores presentes na tabela 21.

<b>Nº de soluções ... Casos De Estudo</b>	<b>... melhores</b>	<b>... obtidas em menos iterações</b>	<b>... obtidas em menos tempo</b>
rede 23 – DL	21	16	10
rede 23 – DN	21	19	8
rede 23A – DL	19	17	7
rede 23A – DN	22	21	9
rede 29 – DL	—	—	—
rede 29 – DN	11	11	3
rede 29A – DL	—	—	—
rede 29A – DN	11	11	6

Tabela 20. Melhorias dos casos de estudo na fase 3, em relação às melhores soluções das fases anteriores.

	<b>Nº de piores cargas iguais</b>	<b>Nº de soluções semelhantes ...</b>	<b>... obtidas em menos iteraões</b>	<b>... obtidas em menos tempo</b>
rede 29 – DL	8	30	21	3
rede 29A – DL	11	21	3	1

Tabela 21. Resultados das redes de 29 nós, para pares de percursos disjuntos nas ligações, em relação às melhores soluções das fases anteriores.

Nas tabelas 20 e 21, na coluna relativa ao número de soluções obtidas em **menos iteraões**, apenas foram consideradas as soluções presentes na coluna à sua esquerda. Da mesma maneira, na coluna relativa ao número de soluções obtidas em **menos tempo**, apenas foram consideradas as soluções presentes na coluna à esquerda desta.

Comparando os resultados da fase 3 com os das fases 1 e 2, conclui-se que, em geral, nesta fase foram alcançados melhores resultados.

Analisando os resultados relativos à rede 23 (tabelas 12 e 13), tanto para pares de percursos disjuntos nas ligações como para pares de percursos disjuntos nos nós, as 21 melhores soluções obtidas são melhores que a “melhor solução” obtida nas fases 1 e 2. Na quinta pior carga, verificou-se uma melhoria de 1 % (Apêndice C – tabelas 40 e 41). Para além disso, mais de metade destas soluções foram obtidas em menos iterações em relação às fases anteriores, sendo que para disjuntividade nas ligações, 10 destas soluções foram obtidas em menos tempo do que nas fases anteriores, enquanto na versão dos nós apenas 8 soluções conseguiram esse feito.

Para a rede 23A (tabelas 14 e 15), tiraram-se as mesmas conclusões que para a rede 23, variando ligeiramente os números de soluções presentes na tabela 20.

Como dito anteriormente, na rede 29 e na rede 29A, para pares de percursos disjuntos nas ligações (tabelas 16 e 18), não se conseguiu melhorar a “melhor solução” das fases 1 e 2, e, por isso, registaram-se os valores presentes na tabela 21. Assim, na rede 29 (Apêndice C – tabela 44), todas as 30 soluções têm as 8 piores cargas iguais à melhor solução das fases anteriores, sendo que 25 são executadas em menos iterações. Já para a rede 29A (Apêndice C – tabela 46), 21 soluções têm as 11 piores cargas iguais à melhor solução das fases anteriores, sendo que 5 delas foram conseguidas em menos iterações. Nestas redes, mas para pares de percursos disjuntos nos nós (tabelas 17 e 19) obteve-se melhorias em 11 soluções, mas em ligações de cargas baixas (Apêndice C – tabelas 45 e 47).

Analisando cuidadosamente os valores das cargas das soluções obtidas, verifica-se que, para os casos de estudo em que foram considerados os pares de percursos disjuntos nas ligações, as soluções divergem mais entre si do que para os casos de estudo para pares de percursos disjuntos nos nós. Este resultado deve-se ao facto de que o espaço de soluções de disjuntividade nas ligações ser maior do que a disjuntividade nos nós, havendo, por isso, maiores alternativas nas soluções.

Como foi dito no Capítulo 3, um percurso disjunto nos nós é ao mesmo tempo disjunto nas ligações:

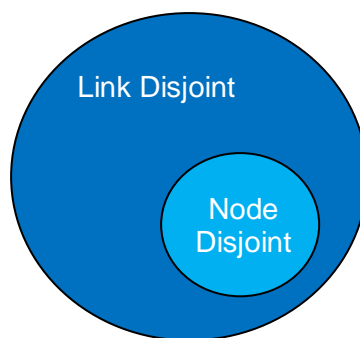


Figura 17. Espaço de soluções possíveis

Os gráficos das figuras 18 e 19 mostram a distribuição das opções utilizadas para variação dos expoentes, na fase 3. Tal como nas tabelas 20 e 21, nas figuras 18 e 19, os casos de estudo em que foram determinados pares de percursos disjuntos nas ligações serão identificados por DL, já os casos em que foram determinados pares de percursos disjuntos nos nós serão identificados por DN.

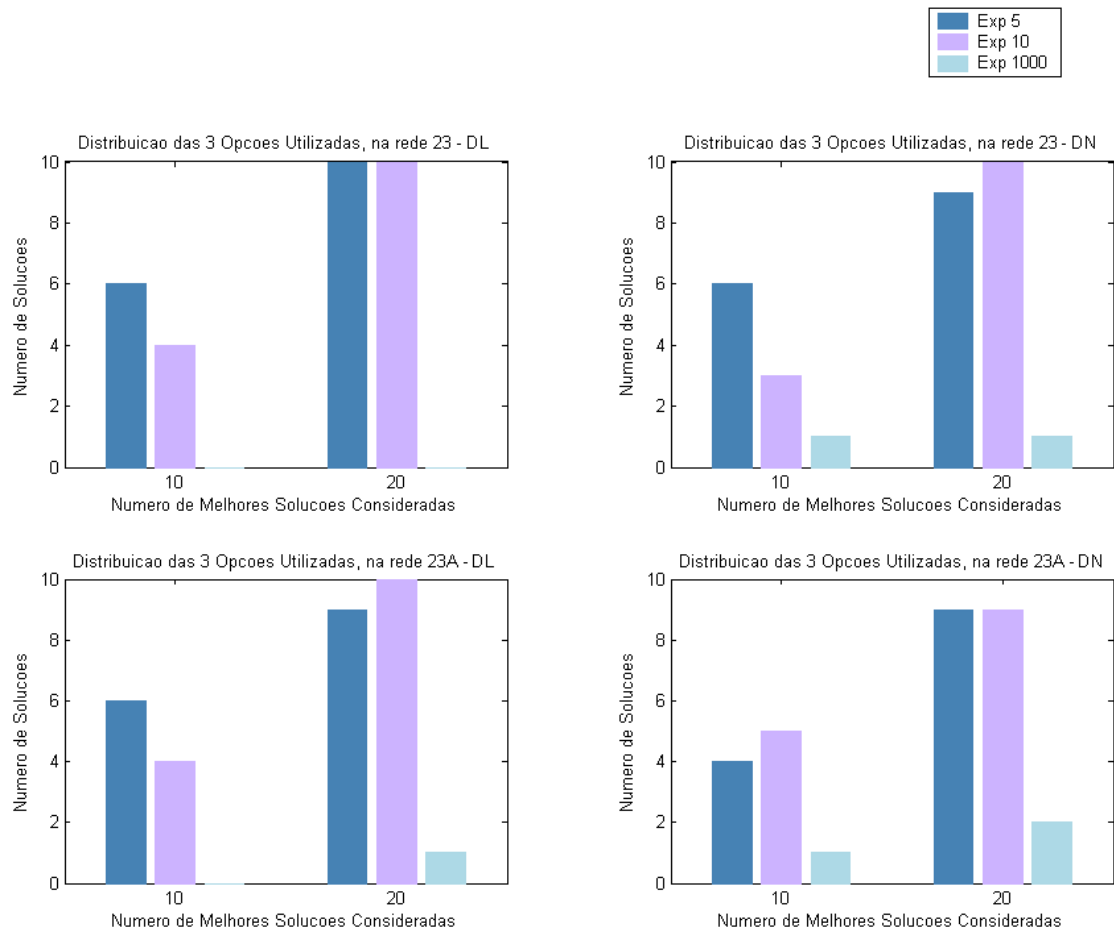


Figura 18. Distribuição de Exp 5, Exp 10 e Exp 1000, para as duas redes de 23 nós.

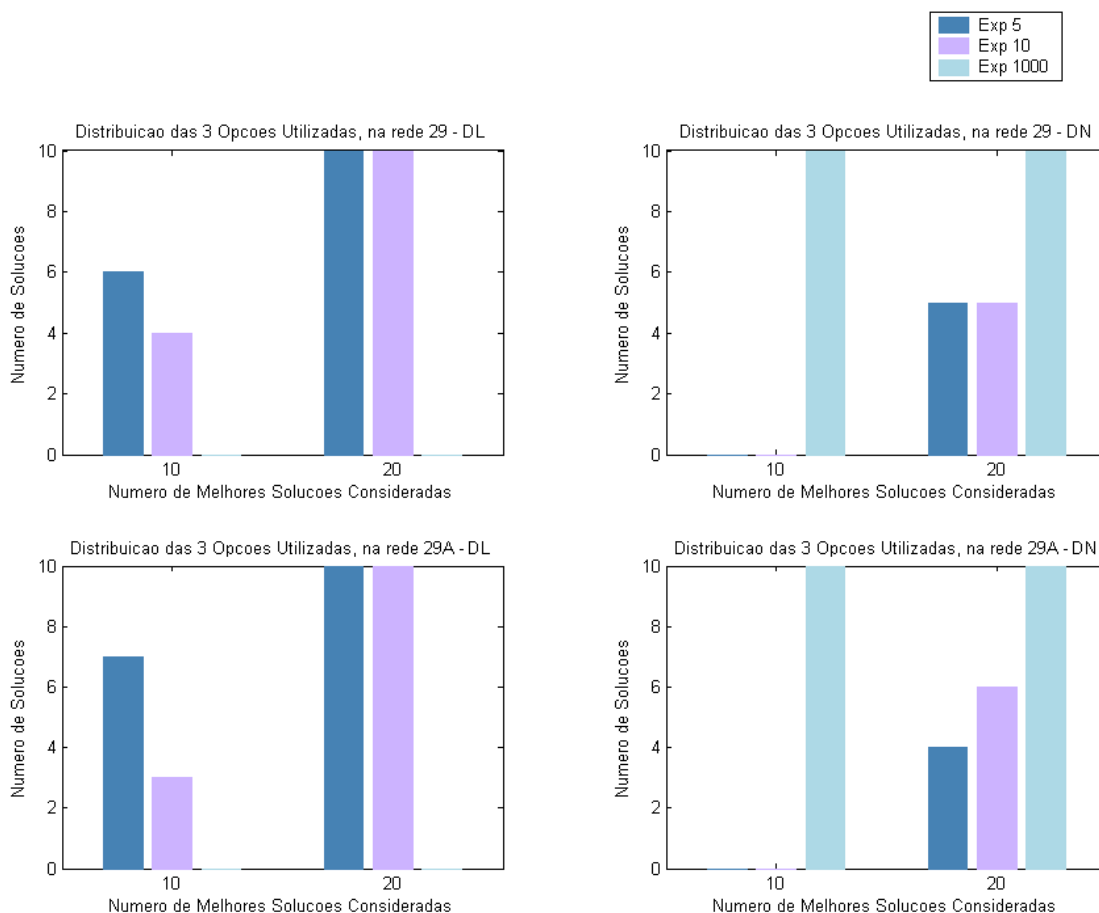


Figura 19. Distribuição de Exp 5, Exp 10 e Exp 1000, para as duas redes de 29 nós.

Na tabela 22, encontram-se contabilizados os casos de estudo em que cada opção (Exp 5, Exp 10 e Exp 1000) foi melhor.

Nº de casos de estudo em que ... Método	Exp 5	Exp 10	Exp 1000
... gera a melhor solução total	4	2	2
... obtém melhor desempenho	5	1	2

Tabela 22. Resultados das 3 opções utilizadas para variação do expoente

Nas figuras 18 e 19, por cada caso de estudo, tem-se o número de soluções originadas por cada opção, nas 10 e nas 20 melhores soluções totais.

Observando as figuras 18 e 19, parece haver uma tendência de que para a versão do algoritmo de pares disjuntos nas ligações, os expoentes pequenos (Exp 5 e Exp 10) apresentam um melhor comportamento. Como seria de esperar, verifica-se que o Exp 1000 obtém os seus melhores resultados, nos mesmos casos de estudo onde nas fases

1 e 2, a melhor solução foi obtida com um expoente elevado, valores que Exp 5 e Exp 10 nunca atingem. Nos restantes casos de estudo, observa-se que tanto Exp 5 como Exp 10 obtêm soluções melhores que nas fases anteriores, no entanto, entre estes, o Exp 5 consegue ter um melhor desempenho (tabela 22), porque como processa menos expoentes, está em mais iterações a fazer a Pesquisa Local Aleatória. Apesar de na fase 2, os expoentes mais baixos obtidos terem, no máximo, valor 2, verificou-se que os limites utilizados em Exp 5 e Exp 10 foram bons, visto que as melhores soluções obtidas na fase 3 terem expoente entre 1 e 10.

Vale a pena referir que as redes com maiores quantidades de tráfego (rede 23A e rede 29A) têm uma distribuição semelhante aos correspondentes casos de estudo originais (rede 23 e rede 29), pois têm a mesma topologia de rede, aumentando apenas a matriz de tráfego presentes nelas.

Em todas as fases estudadas, para todos os casos de estudo, verificou-se que os arcos com as piores cargas tinham sentido *downstream*. Isto justifica-se pelo facto de as maiores larguras de banda do fluxos de tráfego terem os clientes como destino final.

Pela sequência dos resultados obtidos, pode-se concluir que a combinação das duas técnicas de pesquisa utilizadas (Pesquisa Local Aleatória e Pesquisa por Pesos Modificados) resulta em melhorias em relação às “melhores soluções” individuais de cada técnica. Foram verificadas melhorias quer em termos de minimização das piores cargas, como também no número de iterações com que a melhor solução é obtida.



## Capítulo 5

### Conclusões

Neste trabalho foram propostos diferentes algoritmos heurísticos de optimização de redes de telecomunicações, baseados na tecnologia PBB-TE. Os algoritmos foram apresentados reflectindo as diferentes fases do trabalho realizado.

No Capítulo 2, fez-se o estudo das tecnologias que serviram de base a este trabalho. Um dos aspectos principais referidos neste capítulo é o *protection switching* que ocorre no PBB-TE e é a ponte para os algoritmos desenvolvidos.

Os algoritmos desenvolvidos foram descritos no terceiro capítulo, onde são apresentadas as suas fases e as principais rotinas utilizadas. Foram utilizados métodos heurísticos de modo a obter boas soluções em tempo útil. O trabalho passou por 3 fases. Nessas fases, utilizaram-se dois tipos de técnicas de pesquisa para melhorar as soluções. Numa primeira fase, utilizou-se a Pesquisa Local Aleatória, na fase 2 usou-se a Pesquisa por Pesos Modificados e na terceira, e última fase, combinaram-se as duas técnicas anteriores. Cada um dos algoritmos de cada uma destas fases, recebe um ficheiro de entrada com os dados de uma rede e no final devolve um ficheiro com os dados relativos à melhor solução encontrada.

Os resultados computacionais foram apresentados no Capítulo 4, onde estão detalhados os casos de estudo, testes realizados e os seus resultados. Antes de se retirarem os resultados finais, realizaram diversos testes intermédios, na tentativa de melhorar certos procedimentos no algoritmo, melhorando as soluções. Para cada caso de estudo, para cada fase do algoritmo, retiraram-se resultados. A sua análise cuidada, permite concluir que na fase 3, onde se conjugam as duas técnicas de pesquisa referidas anteriormente, obtém-se melhores resultados, pois praticamente em todos os casos de estudo, as soluções das fases 1 e 2, são melhoradas. Assim, conclui-se que a minimização das piores cargas de uma rede pode ser optimizada, recorrendo à combinação de duas técnicas, uma baseada em trocas de fluxos e outra baseada em modificação dos pesos dos arcos da rede.

Apesar destes resultados, as melhorias das soluções seriam mais satisfatórias se a topologia das redes fosse ligeiramente alterada. Esta alteração consistiria em uniformizar, por *switch*, os percursos disjuntos possíveis. Assim, todos os *switches* teriam um número de percursos alternativos semelhantes, fazendo com que a minimização das piores cargas fosse mais pronunciada.

Caso o PBB-TE ou outra tecnologia semelhante venha ser utilizada em larga escala, neste trabalho estão um conjunto de testes que podem ajudar na implementação dos algoritmos que suportem essas tecnologias.

## Apêndice A – Exemplo de Utilização

Um exemplo completo de um ficheiro de entrada correcto é o seguinte:

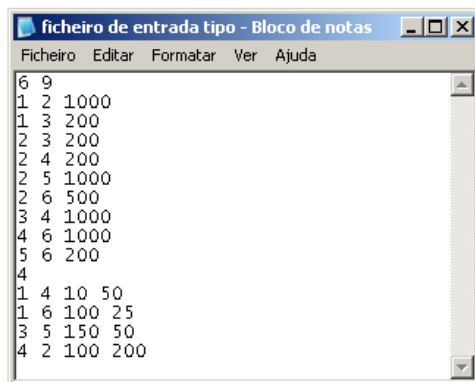


Figura 20. Exemplo de um ficheiro de entrada

Apesar de, na imagem anterior, as ligações estarem ordenadas pelo vértice origem, não é imposto nenhum tipo de ordenação. O utilizador deve respeitar que todos os vértices devem ter pelo menos duas ligações, para possibilitar o cálculo dos pares de percursos disjuntos. A figura 21 é a esquematização da rede relativa ao ficheiro da figura 20:

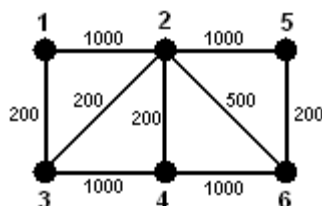


Figura 21. Rede esquematizada do exemplo anterior

Analisando a figura 21., é possível verificar que junto de cada nó está o respectivo número do vértice e em cada ligação a respectiva capacidade de transmissão.

Depois de se introduzir as ligações da rede, tem-se que colocar na linha seguinte (linha número <número de ligações> + 2) o número de fluxos que se pretende fazer encaminhar.

Seguidamente, introduz-se cada fluxo com a respectiva característica, dada por vértice origem, vértice destino, largura de banda da origem para o destino e do destino para a origem.

Vale a pena referir que o número de linhas escritas no ficheiro de entrada é igual à soma de <número de ligações> com <número de fluxos> mais dois, este último parâmetro diz respeito às linhas <número de vértices> <número de ligações> e <número de fluxos>.

Em seguida, é pedido ao utilizador:

- A. o nome do ficheiro de saída onde pretende gravar os resultados finais;
- B. a fase do algoritmo que quer testar;
- C. o número de iterações;
- D. a determinação de par de percursos disjuntos nas ligações ou par de percursos disjuntos nos nós.

Em B., caso o utilizador seleccione a Pesquisa por Pesos Modificados ser-lhe-á pedido que introduza o valor *delta* desejado, caso a opção seja o algoritmo combinando a Pesquisa Local Aleatória com a Pesquisa por Pesos Modificados então ser-lhe-á pedido que introduza o tipo de variação de expoente que pretende.

O conteúdo da matriz auxiliar de capacidades é:

0.0	1000.0	200.0	0.0	0.0	0.0
1000.0	0.0	200.0	200.0	1000.0	500.0
200.0	200.0	0.0	1000.0	0.0	0.0
0.0	200.0	1000.0	0.0	0.0	1000.0
0.0	1000.0	0.0	0.0	0.0	200.0
0.0	500.0	0.0	1000.0	200.0	0.0

Figura 22. Matriz com a capacidade de cada ligação do exemplo considerado

Em seguida, são apresentados a ordenação dos arcos (tabela 23), assim como também a disposição do *array index* (tabela 24) referido no subcapítulo 3.5.3.5 aquando da explicação da FSF, segundo o exemplo dado na figura 20:

Número do Arco	Vértice Origem	Vértice Destino	Capacidade do Arco
1	1	2	1000.000
2	1	3	200.000
3	2	1	1000.000
4	2	3	200.000
5	2	4	200.000
6	2	5	1000.000
7	2	6	500.000
8	3	1	200.000
9	3	2	200.000
10	3	4	1000.000
11	4	2	200.000
12	4	3	1000.000
13	4	6	1000.000
14	5	2	1000.000
15	5	6	200.000
16	6	2	500.000
17	6	4	1000.000
18	6	5	200.000

Tabela 23. Lista dos arcos ordenados na Forward Star Form

<b>Vértice i</b>	<b><i>index(i)</i></b>
1	1
2	3
3	8
4	11
5	14
6	16
(7)	19

Tabela 24. Ponteiro para cada nó na Forward Star Form

```

formato ficheiro de saída - Bloco de notas
Ficheiro Editar Formatar Ver Ajuda

##### Resultados Finais #####
Tempo total de execução:      0.00
Instante da melhor solução:   0.00
Expoente da melhor solução:   1.0
N. melhorias da melhor solução: 0
Iteracoes Totais:            2000
Iteracao da melhor solucao:   0
Pior carga da melhor solucao: 1.0000

Estado de cada ligacao da melhor solucao:
1      2      0.1100
1      3      0.5500
2      1      0.0750
2      3      0.2500
2      4      1.0000
2      5      0.1500
2      6      0.6200
3      1      0.3750
3      2      0.7500
3      4      0.2600
4      2      0.5000
4      3      0.1250
4      6      0.4000
5      2      0.0500
5      6      0.2500
6      2      0.3500
6      4      0.2850
6      5      0.7500

Sequencia dos percursos de servico e proteccao da melhor solucao (versao resumida):

4      6      2
4      2

3      2      5
3      4      6      5

1      2      6
1      3      4      6

1      3      4
1      2      6      4

Sequencia dos percursos de servico e proteccao da melhor solucao (versao detalhada):

4      2      100.0000      200.0000
4 ( 0.4000 - 0.2850 )      6 ( 0.3500 - 0.6200 )      2
4 ( 0.5000 - 1.0000 )      2

3      5      150.0000      50.0000
3 ( 0.7500 - 0.2500 )      2 ( 0.1500 - 0.0500 )      5
3 ( 0.2600 - 0.1250 )      4 ( 0.4000 - 0.2850 )      6 ( 0.7500 - 0.2500 )      5

1      6      100.0000      25.0000
1 ( 0.1100 - 0.0750 )      2 ( 0.6200 - 0.3500 )      6
1 ( 0.5500 - 0.3750 )      3 ( 0.2600 - 0.1250 )      4 ( 0.4000 - 0.2850 )      6

1      4      10.0000      50.0000
1 ( 0.5500 - 0.3750 )      3 ( 0.2600 - 0.1250 )      4
1 ( 0.1100 - 0.0750 )      2 ( 0.6200 - 0.3500 )      6 ( 0.2850 - 0.4000 )      4

Cargas ordenadas:
1.0000
0.7500
0.7500
0.6200
0.5500
0.5000
0.4000
0.3750
0.3500
0.2850
0.2600
0.2500
0.2500
0.1500
0.1250
0.1100
0.0750
0.0500

```

Figura 23. Exemplo de um ficheiro de saída

## Apêndice B – Formato dos Dados e dos Resultados

Depois de correr o ficheiro executável do programa, é pedido ao utilizador que introduza o nome do ficheiro de entrada que tenha como seu conteúdo a topologia de uma rede e os fluxos que pretende encaminhar. O ficheiro deve respeitar o seguinte formato:

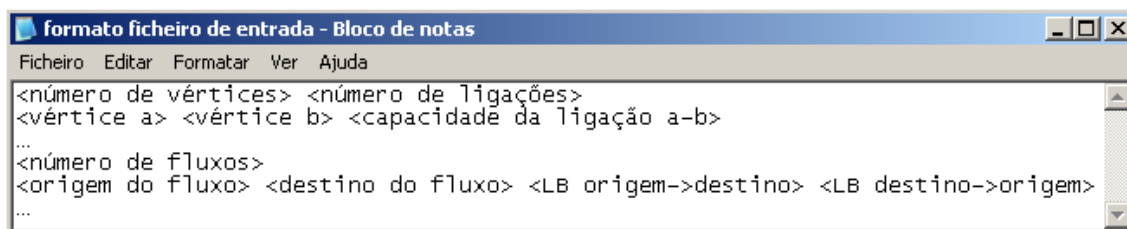


Figura 24. Formatação que deve ter o ficheiro de entrada

Na primeira linha do ficheiro deve estar presente o número de vértices e de ligações da rede. Como uma parte integrante do algoritmo é a determinação de um par de percursos disjuntos nos nós e/ou nas ligações, cabe ao utilizador, verificar se a rede em estudo tem esta característica para todos os nós. Assim, facilmente se conclui que os valores mínimos para os vértices e para as ligações são 3. Assim, caso não haja nenhuma ligação com carga superior a 100%, garante-se sempre os dois percursos pretendidos.

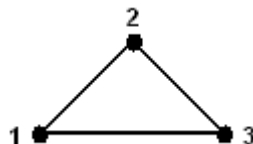
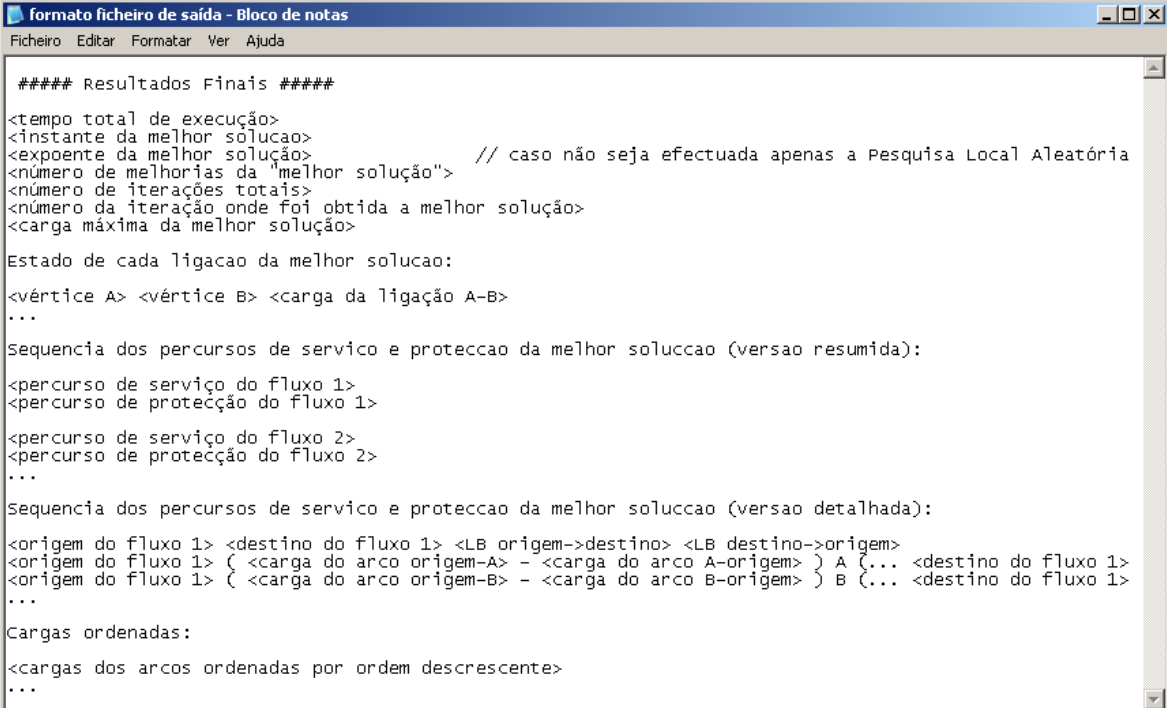


Figura 25. Configuração mínima que a rede pode apresentar

Da segunda linha até à linha <número de ligações> + 1, devem estar dispostas todas as ligações da rede. Cada ligação deve conter os dois vértices que ela une e a sua capacidade. A ordem com que os vértices são introduzidos não é relevante.

Não esquecer que o arco (i, j) e o arco (j, i) representam a mesma ligação, por isso o <número de ligações> é igual a metade do número total de arcos. Assim, ao introduzir os dados da rede no ficheiro de texto, deve ter-se o cuidado de não repetir ligações.

Depois de efectuados todos os testes, resta guardar os resultados no ficheiro de saída definido no início do programa. O formato do ficheiro de saída é o seguinte:



```
##### Resultados Finais #####
<tempo total de execução>
<instante da melhor solucao>
<expoente da melhor solução> // caso não seja efectuada apenas a Pesquisa Local Aleatória
<número de melhorias da "melhor solução">
<número de iterações totais>
<número da iteração onde foi obtida a melhor solução>
<carga máxima da melhor solução>

Estado de cada ligacao da melhor solucao:
<vértice A> <vértice B> <carga da ligação A-B>
...

Sequencia dos percursos de servico e proteccao da melhor solucao (versao resumida):
<percurso de serviço do fluxo 1>
<percurso de protecção do fluxo 1>
...
<percurso de serviço do fluxo 2>
<percurso de protecção do fluxo 2>
...

Sequencia dos percursos de servico e proteccao da melhor solucao (versao detalhada):
<origem do fluxo 1> <destino do fluxo 1> <LB origem->destino> <LB destino->origem>
<origem do fluxo 1> ( <carga do arco origem-A> - <carga do arco A-origem> ) A (... <destino do fluxo 1>
<origem do fluxo 1> ( <carga do arco origem-B> - <carga do arco B-origem> ) B (... <destino do fluxo 1>
...

Cargas ordenadas:
<cargas dos arcos ordenadas por ordem decrescente>
...
```

Figura 26. Formato do ficheiro de saída



## Apêndice C – Resultados

Em seguida encontram-se as soluções iniciais dos 10 casos de estudo utilizados:

Solução do Algoritmo Construtivo					
Disjunto nas Ligações			Disjunto nos nós		
Origem	Destino	Carga	Origem	Destino	Carga
11	3	34,0%	11	3	34,0%
12	3	34,0%	12	3	34,0%
12	4	34,0%	12	4	34,0%
13	4	34,0%	13	4	34,0%
13	6	33,0%	13	6	33,0%
14	6	33,0%	14	6	33,0%
14	8	30,0%	14	8	30,0%
15	8	30,0%	15	8	30,0%
12	5	26,0%	12	5	26,0%
13	5	26,0%	13	5	26,0%
14	9	24,0%	14	9	24,0%
15	9	24,0%	15	9	24,0%
11	1	23,0%	11	1	23,0%
12	1	23,0%	12	1	23,0%
13	7	23,0%	13	7	23,0%
14	7	23,0%	14	7	23,0%
4	12	21,6%	15	10	23,0%
4	13	21,6%	4	12	21,6%
6	13	21,4%	4	13	21,6%
6	14	21,4%	6	13	21,4%
14	10	20,0%	6	14	21,4%
15	10	20,0%	14	10	20,5%
11	2	19,0%	11	2	19,0%
12	2	19,0%	12	2	19,0%
8	14	18,4%	16	13	18,5%
8	15	18,4%	8	14	18,4%
16	13	18,1%	8	15	18,4%
16	11	16,4%	16	11	17,1%
3	11	15,8%	3	11	15,8%
3	12	15,8%	3	12	15,8%
9	14	15,8%	9	14	15,8%
9	15	15,8%	9	15	15,8%
5	12	14,4%	10	14	14,8%
5	13	14,4%	5	12	14,4%
13	12	12,8%	5	13	14,4%
16	15	12,7%	16	15	13,8%
10	14	11,8%	10	15	12,3%
10	15	11,8%	13	12	12,1%
7	13	11,4%	7	13	11,4%
7	14	11,4%	7	14	11,4%
11	16	9,9%	11	16	10,6%
1	11	9,8%	11	12	10,5%
1	12	9,8%	1	11	9,8%
11	12	9,8%	1	12	9,8%
13	14	9,3%	2	11	9,1%
2	11	9,1%	2	12	9,1%
2	12	9,1%	15	16	8,7%
12	13	8,7%	13	14	8,2%
15	16	7,6%	12	11	8,1%
12	11	7,4%	12	13	8,0%

15	14	6,9%	15	14	6,9%
14	13	6,5%	13	16	6,9%
13	16	6,5%	14	13	5,4%
14	15	4,6%	14	15	4,8%

Tabela 25. Cargas dos arcos da solução inicial da rede 16, usando percursos disjuntos nas ligações e nos nós

Solução do Algoritmo Construtivo					
Disjunto nas Ligações			Disjunto nos nós		
Origem	Destino	Carga	Origem	Destino	Carga
23	19	39,1%	23	19	39,6%
19	7	34,0%	19	7	34,0%
20	7	34,0%	20	7	34,0%
20	11	34,0%	19	9	33,0%
21	11	34,0%	20	9	33,0%
19	9	33,0%	21	11	33,0%
20	9	33,0%	22	15	33,0%
22	15	33,0%	21	14	31,0%
21	14	31,0%	22	14	31,0%
22	14	31,0%	23	22	29,7%
19	8	29,0%	19	8	29,0%
20	8	29,0%	20	8	29,0%
21	12	29,0%	21	12	29,0%
22	12	29,0%	22	12	29,0%
17	2	28,0%	17	2	28,0%
18	2	28,0%	18	2	28,0%
21	15	28,0%	20	11	28,0%
17	4	27,0%	21	15	28,0%
18	4	27,0%	22	16	28,0%
19	6	27,0%	17	4	27,0%
23	22	26,8%	18	4	27,0%
17	1	26,0%	17	1	26,0%
17	3	26,0%	17	3	26,0%
18	1	26,0%	18	1	26,0%
18	3	26,0%	18	3	26,0%
19	10	25,0%	19	6	25,0%
20	10	25,0%	19	10	25,0%
22	16	25,0%	20	10	25,0%
18	6	24,5%	17	5	23,0%
17	5	23,0%	18	5	23,0%
18	5	23,0%	21	13	23,0%
21	13	23,0%	21	16	23,0%
22	13	23,0%	22	13	23,0%
21	16	22,5%	18	6	22,5%
19	20	20,1%	20	21	19,6%
20	21	18,7%	23	17	19,1%
23	17	18,3%	21	20	18,6%
21	20	18,2%	19	20	18,0%
15	21	17,3%	15	21	17,3%
19	18	17,2%	19	18	16,6%
11	20	16,6%	22	21	16,1%
11	21	16,6%	9	19	15,7%
9	19	15,7%	9	20	15,7%
9	20	15,7%	11	20	15,6%
22	21	14,3%	16	21	14,8%
6	18	13,8%	19	23	13,4%
4	17	12,9%	4	17	12,9%
4	18	12,9%	4	18	12,9%
19	23	12,9%	22	23	12,8%
7	19	12,4%	7	19	12,4%
7	20	12,4%	7	20	12,4%
12	21	12,4%	12	21	12,4%
12	22	12,4%	12	22	12,4%
15	22	12,3%	15	22	12,3%
16	21	11,8%	6	18	11,8%

6	19	11,3%	14	21	11,2%
14	21	11,2%	14	22	11,2%
14	22	11,2%	11	21	10,6%
20	19	10,5%	10	19	10,1%
10	19	10,1%	10	20	10,1%
10	20	10,1%	8	19	9,9%
22	23	9,9%	8	20	9,9%
8	19	9,9%	2	17	9,8%
8	20	9,9%	2	18	9,8%
2	17	9,8%	5	17	9,8%
2	18	9,8%	5	18	9,8%
5	17	9,8%	13	21	9,8%
5	18	9,8%	13	22	9,8%
13	21	9,8%	16	22	9,8%
13	22	9,8%	17	23	9,8%
1	17	9,5%	1	17	9,5%
1	18	9,5%	1	18	9,5%
18	19	9,4%	6	19	9,3%
16	22	9,3%	18	19	8,8%
17	23	9,0%	20	19	8,4%
3	17	7,8%	21	22	8,1%
3	18	7,8%	3	17	7,8%
17	18	6,9%	3	18	7,8%
21	22	6,0%	17	18	6,9%
18	17	5,6%	18	17	5,6%

Tabela 26. Cargas dos arcos da solução inicial da rede 23, usando percursos disjuntos nas ligações e nos nós

Solução do Algoritmo Construtivo					
Disjunto nas Ligações			Disjunto nos nós		
Origem	Destino	Carga	Origem	Destino	Carga
23	19	45,7%	23	19	45,2%
19	7	41,0%	19	9	39,0%
20	7	41,0%	20	9	39,0%
19	9	39,0%	22	13	39,0%
20	9	39,0%	17	3	37,0%
17	3	37,0%	18	3	37,0%
18	3	37,0%	21	11	37,0%
20	11	36,0%	21	13	36,5%
21	11	36,0%	19	7	36,0%
21	13	36,0%	20	7	36,0%
22	13	36,0%	22	15	35,0%
22	15	35,0%	20	11	34,5%
17	1	34,0%	23	22	34,5%
18	1	34,0%	17	1	34,0%
19	8	34,0%	18	1	34,0%
20	8	34,0%	22	14	34,0%
17	2	33,0%	17	2	33,0%
17	4	33,0%	17	4	33,0%
18	2	33,0%	18	2	33,0%
18	4	33,0%	18	4	33,0%
23	22	31,1%	22	12	32,0%
21	14	31,0%	21	14	31,5%
22	14	31,0%	21	15	30,0%
21	15	30,0%	22	16	30,0%
17	5	29,0%	21	12	29,5%
18	5	29,0%	17	5	29,0%
18	6	29,0%	18	5	29,0%
19	6	29,0%	19	8	29,0%
21	12	29,0%	20	8	29,0%
22	12	29,0%	21	16	27,5%
19	10	27,0%	18	6	27,0%
20	10	27,0%	19	6	27,0%
21	16	27,0%	19	10	27,0%
22	16	27,0%	20	10	27,0%

19	20	23,4%	23	17	23,7%
20	21	22,7%	20	21	23,7%
23	17	22,4%	21	20	22,6%
19	18	22,1%	19	20	21,8%
21	20	21,9%	19	18	21,0%
7	19	19,4%	11	20	19,6%
7	20	19,4%	15	21	19,3%
15	21	19,3%	22	21	19,3%
11	20	18,6%	13	21	18,3%
11	21	18,6%	11	21	17,1%
22	21	17,9%	16	21	16,8%
9	19	16,7%	9	19	16,7%
9	20	16,7%	9	20	16,7%
19	23	16,0%	22	23	16,4%
6	18	15,8%	13	22	15,8%
6	19	15,8%	19	23	15,5%
13	21	15,3%	12	21	15,4%
13	22	15,3%	2	17	14,8%
8	19	14,9%	2	18	14,8%
8	20	14,9%	7	19	14,4%
2	17	14,8%	7	20	14,4%
2	18	14,8%	15	22	14,3%
15	22	14,3%	16	22	14,3%
4	17	13,9%	14	21	14,2%
4	18	13,9%	4	17	13,9%
16	21	13,8%	4	18	13,9%
16	22	13,8%	6	18	13,8%
20	19	13,1%	6	19	13,8%
22	23	13,0%	12	22	12,9%
1	17	12,5%	1	17	12,5%
1	18	12,5%	1	18	12,5%
12	21	12,4%	17	23	12,1%
12	22	12,4%	10	19	12,1%
10	19	12,1%	10	20	12,1%
10	20	12,1%	14	22	11,7%
18	19	11,8%	20	19	11,5%
3	17	11,3%	3	17	11,3%
3	18	11,3%	3	18	11,3%
14	21	11,2%	21	22	11,3%
14	22	11,2%	5	17	10,8%
17	23	10,8%	5	18	10,8%
5	17	10,8%	18	19	10,7%
5	18	10,8%	8	19	9,9%
21	22	8,9%	8	20	9,9%
17	18	8,4%	17	18	8,9%
18	17	7,1%	18	17	7,6%

Tabela 27. Cargas dos arcos da solução inicial da rede 23A, usando percursos disjuntos nas ligações e nos nós

Solução do Algoritmo Construtivo					
Disjunto nas Ligações			Disjunto nos nós		
Origem	Destino	Carga	Origem	Destino	Carga
22	6	42,0%	22	6	42,0%
23	6	42,0%	23	6	42,0%
23	7	36,0%	23	7	36,0%
24	7	36,0%	24	7	36,0%
24	10	35,0%	24	10	35,0%
25	10	35,0%	25	10	35,0%
29	26	33,7%	29	26	33,1%
21	2	33,0%	21	2	33,0%
26	15	32,0%	26	15	32,0%
27	15	32,0%	27	15	32,0%
29	23	31,9%	29	23	31,4%
22	2	30,5%	22	2	30,5%
21	5	30,0%	21	5	30,0%

22	5	30,0%	22	5	30,0%
25	13	29,0%	25	13	29,0%
26	13	29,0%	26	13	29,0%
23	8	28,0%	23	8	28,0%
23	9	28,0%	23	9	28,0%
24	8	28,0%	24	8	28,0%
24	9	28,0%	24	9	28,0%
28	16	28,0%	25	11	28,0%
28	20	28,0%	28	16	28,0%
21	3	26,0%	28	20	28,0%
22	3	26,0%	21	3	26,0%
26	14	26,0%	22	3	26,0%
27	16	25,5%	26	14	26,0%
27	20	25,5%	27	16	25,5%
21	1	25,0%	27	20	25,5%
22	1	25,0%	21	1	25,0%
25	11	25,0%	22	1	25,0%
26	12	25,0%	26	12	25,0%
24	11	22,5%	24	11	23,0%
21	4	22,0%	21	4	22,0%
22	4	22,0%	22	4	22,0%
6	22	21,3%	6	22	21,3%
6	23	21,3%	6	23	21,3%
29	21	21,2%	29	21	21,2%
25	14	21,0%	25	14	21,0%
28	19	21,0%	28	19	21,0%
29	28	20,8%	25	12	20,0%
25	12	20,0%	28	17	20,0%
28	17	20,0%	26	25	19,6%
23	22	18,7%	27	19	18,5%
27	19	18,5%	23	24	18,4%
27	18	18,0%	29	28	18,1%
28	18	18,0%	23	22	18,0%
26	25	17,9%	27	18	18,0%
10	24	17,7%	28	18	18,0%
10	25	17,7%	10	24	17,7%
27	17	17,5%	10	25	17,7%
5	21	16,0%	27	17	17,5%
5	22	16,0%	25	24	16,7%
26	27	15,4%	24	25	16,1%
23	24	15,1%	5	21	16,0%
2	22	14,8%	5	22	16,0%
15	26	14,6%	26	27	14,9%
15	27	14,6%	2	22	14,8%
12	25	14,4%	15	26	14,6%
8	23	14,0%	15	27	14,6%
8	24	14,0%	12	25	14,4%
7	23	13,6%	8	23	14,0%
7	24	13,6%	8	24	14,0%
14	25	12,9%	7	23	13,6%
26	29	12,8%	7	24	13,6%
2	21	12,3%	11	24	13,2%
23	29	12,1%	14	25	12,9%
3	21	12,0%	2	21	12,3%
3	22	12,0%	25	26	12,1%
22	23	12,0%	3	21	12,0%
25	24	11,8%	3	22	12,0%
9	23	11,4%	21	29	11,9%
9	24	11,4%	26	29	11,7%
21	29	11,4%	9	23	11,4%
1	21	11,0%	9	24	11,4%
1	22	11,0%	1	21	11,0%
13	25	10,8%	1	22	11,0%
13	26	10,8%	24	23	11,0%
27	26	10,7%	13	25	10,8%
16	27	10,6%	13	26	10,8%
11	24	10,2%	22	23	10,8%
28	29	9,9%	16	27	10,6%
4	21	9,7%	4	21	9,7%

4	22	9,7%	4	22	9,7%
28	27	9,7%	19	27	9,5%
19	27	9,5%	28	29	9,4%
24	23	9,5%	12	26	9,4%
12	26	9,4%	23	29	9,4%
24	25	9,1%	18	27	9,0%
18	27	9,0%	18	28	9,0%
18	28	9,0%	20	27	8,9%
20	27	8,9%	11	25	8,2%
21	22	8,8%	16	28	8,1%
25	26	8,6%	21	22	8,0%
16	28	8,1%	27	26	8,0%
14	26	7,9%	14	26	7,9%
11	25	7,7%	17	27	7,7%
17	27	7,7%	19	28	7,0%
19	28	7,0%	28	27	7,0%
27	28	6,7%	20	28	6,4%
22	21	6,5%	27	28	6,3%
20	28	6,4%	22	21	6,2%
17	28	5,2%	17	28	5,2%

Tabela 28. Cargas dos arcos da solução inicial da rede 29, usando percursos disjuntos nas ligações e nos nós

Solução do Algoritmo Construtivo					
Disjunto nas Ligações			Disjunto nos nós		
Origem	Destino	Carga	Origem	Destino	Carga
22	6	59,0%	22	6	54,0%
23	6	59,0%	23	6	54,0%
23	9	49,0%	23	7	47,0%
24	9	49,0%	24	7	47,0%
23	7	47,0%	21	3	44,0%
24	7	47,0%	22	3	44,0%
21	3	44,0%	23	9	44,0%
22	3	44,0%	24	9	44,0%
24	10	44,0%	24	10	44,0%
25	10	44,0%	25	10	44,0%
26	13	42,0%	26	13	42,0%
21	2	41,0%	29	23	41,2%
28	16	40,0%	21	2	41,0%
25	13	39,5%	28	16	40,0%
25	12	39,0%	29	26	39,6%
26	12	39,0%	25	13	39,5%
29	23	39,0%	22	2	38,5%
29	26	38,6%	26	15	38,0%
22	2	38,5%	27	15	38,0%
26	15	38,0%	27	16	37,5%
27	15	38,0%	21	1	37,0%
27	16	37,5%	22	1	37,0%
21	1	37,0%	26	12	37,0%
22	1	37,0%	26	14	36,0%
26	14	37,0%	25	12	34,5%
28	20	34,0%	28	20	34,0%
6	22	33,3%	25	14	33,5%
6	23	33,3%	21	4	33,0%
21	4	33,0%	22	4	33,0%
22	4	33,0%	27	20	31,5%
27	20	31,5%	24	11	31,0%
24	11	31,0%	25	11	31,0%
25	11	31,0%	21	5	30,0%
21	5	30,0%	22	5	30,0%
22	5	30,0%	23	8	30,0%
23	8	30,0%	24	8	30,0%
24	8	30,0%	6	22	28,3%
25	14	29,5%	6	23	28,3%

27	17	28,0%	27	17	28,0%
28	17	28,0%	28	17	28,0%
29	21	27,1%	29	21	27,4%
29	28	26,9%	10	24	26,7%
10	24	26,7%	10	25	26,7%
10	25	26,7%	26	25	25,1%
28	19	25,0%	28	19	25,0%
9	23	24,9%	7	23	24,6%
9	24	24,9%	7	24	24,6%
26	25	24,7%	23	24	24,5%
7	23	24,6%	25	24	23,5%
7	24	24,6%	23	22	23,0%
14	25	23,9%	29	28	23,0%
23	22	23,8%	14	25	22,9%
12	25	23,4%	24	25	22,7%
12	26	23,4%	16	27	22,6%
16	27	22,6%	3	21	22,5%
3	21	22,5%	3	22	22,5%
3	22	22,5%	27	19	22,5%
27	19	22,5%	12	25	21,4%
1	21	20,5%	1	21	20,5%
1	22	20,5%	1	22	20,5%
16	28	20,1%	14	26	20,4%
26	27	19,9%	16	28	20,1%
23	24	19,1%	9	23	19,9%
13	25	18,8%	9	24	19,9%
27	18	18,0%	26	27	19,2%
28	18	18,0%	12	26	18,9%
2	22	17,8%	13	25	18,8%
25	24	16,9%	27	18	18,0%
14	26	16,4%	28	18	18,0%
13	26	16,3%	2	22	17,8%
11	24	16,2%	25	26	16,6%
11	25	16,2%	13	26	16,3%
5	21	16,0%	11	24	16,2%
5	22	16,0%	11	25	16,2%
8	23	16,0%	24	23	16,1%
8	24	16,0%	5	21	16,0%
23	29	16,0%	5	22	16,0%
4	21	15,7%	8	23	16,0%
4	22	15,7%	8	24	16,0%
17	27	15,7%	4	21	15,7%
17	28	15,7%	4	22	15,7%
27	26	15,7%	17	27	15,7%
15	26	15,6%	17	28	15,7%
15	27	15,6%	15	26	15,6%
2	21	15,3%	15	27	15,6%
22	23	15,2%	21	29	15,6%
26	29	15,1%	26	29	15,5%
21	29	14,6%	2	21	15,3%
28	29	14,0%	23	29	14,9%
25	26	13,8%	22	23	13,8%
19	27	13,5%	19	27	13,5%
24	25	13,4%	28	29	13,3%
24	23	13,3%	27	26	11,8%
28	27	12,8%	19	28	11,0%
21	22	11,2%	20	27	9,9%
19	28	11,0%	21	22	9,7%
20	27	9,9%	18	27	9,0%
18	27	9,0%	18	28	9,0%
18	28	9,0%	28	27	8,9%
22	21	8,2%	27	28	7,4%
27	28	8,1%	20	28	7,4%
20	28	7,4%	22	21	7,4%

Tabela 29. Cargas dos arcos da solução inicial da rede 29A, usando percursos disjuntos nas ligações e nos nós

[illegible]

Tabela 30. Estado das 16 piores cargas das soluções relativas à rede 16, para pares de percursos disjuntos nas ligações – Fases 1 e 2

[illegible]

Tabela 31. Estado das 16 piores cargas das soluções relativas à rede 16, para pares de percursos disjuntos nos nós – Fases 1 e 2



0,34	0,34	0,33	0,33	0,33	0,33	0,324	0,32	0,32	0,32	0,32	0,31	0,31	0,29
0,34	0,34	0,33	0,33	0,33	0,33	0,327	0,32	0,32	0,32	0,32	0,31	0,31	0,29
0,34	0,34	0,33	0,33	0,33	0,33	0,327	0,32	0,32	0,32	0,32	0,31	0,31	0,29
0,34	0,34	0,33	0,33	0,33	0,33	0,327	0,32	0,32	0,32	0,32	0,31	0,31	0,29
0,34	0,34	0,33	0,33	0,33	0,33	0,328	0,32	0,32	0,32	0,32	0,31	0,31	0,29
0,34	0,34	0,33	0,33	0,33	0,33	0,328	0,32	0,32	0,32	0,32	0,31	0,31	0,29
0,34	0,34	0,33	0,33	0,33	0,33	0,329	0,32	0,32	0,31	0,31	0,31	0,31	0,29
0,34	0,34	0,33	0,33	0,33	0,33	0,33	0,32	0,32	0,32	0,32	0,31	0,31	0,29
0,34	0,34	0,331	0,33	0,33	0,33	0,33	0,32	0,32	0,32	0,32	0,31	0,31	0,29
0,34	0,34	0,331	0,33	0,33	0,33	0,33	0,32	0,32	0,32	0,32	0,31	0,31	0,29
0,34	0,34	0,334	0,33	0,33	0,33	0,33	0,33	0,322	0,32	0,32	0,31	0,31	0,29
0,34	0,34	0,337	0,33	0,33	0,33	0,33	0,32	0,32	0,32	0,32	0,31	0,31	0,29
0,34	0,34	0,337	0,33	0,33	0,33	0,33	0,32	0,32	0,32	0,32	0,31	0,31	0,29
0,34	0,34	0,338	0,33	0,33	0,33	0,33	0,32	0,32	0,32	0,32	0,31	0,31	0,29
0,34	0,34	0,34	0,33	0,33	0,33	0,33	0,32	0,32	0,31	0,31	0,29	0,29	0,29
0,34	0,34	0,34	0,34	0,338	0,33	0,33	0,33	0,33	0,33	0,32	0,32	0,31	0,31
0,342	0,34	0,34	0,34	0,34	0,33	0,33	0,33	0,33	0,31	0,31	0,29	0,29	0,29
0,344	0,34	0,34	0,34	0,34	0,33	0,33	0,33	0,33	0,31	0,31	0,29	0,29	0,29
0,344	0,34	0,34	0,34	0,34	0,33	0,33	0,33	0,33	0,31	0,31	0,31	0,294	0,29
0,345	0,34	0,34	0,34	0,34	0,33	0,33	0,33	0,33	0,31	0,31	0,29	0,29	0,29
0,361	0,34	0,34	0,34	0,34	0,33	0,33	0,33	0,31	0,31	0,31	0,29	0,29	0,29
0,37	0,37	0,345	0,345	0,33	0,33	0,33	0,33	0,32	0,32	0,31	0,31	0,31	0,295

Tabela 32. Estado das 14 piores cargas das soluções relativas à rede 23, para pares de percursos disjuntos nas ligações – Fases 1 e 2

0,34	0,34	0,33	0,33	0,33	0,33	0,324	0,31	0,31	0,298	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,33	0,33	0,324	0,31	0,31	0,3	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,33	0,33	0,325	0,31	0,31	0,29	0,29	0,29	0,29	0,287
0,34	0,34	0,33	0,33	0,33	0,33	0,325	0,31	0,31	0,29	0,29	0,29	0,29	0,287
0,34	0,34	0,33	0,33	0,33	0,33	0,325	0,31	0,31	0,29	0,29	0,29	0,29	0,288
0,34	0,34	0,33	0,33	0,33	0,33	0,326	0,31	0,31	0,29	0,29	0,29	0,29	0,287
0,34	0,34	0,33	0,33	0,33	0,33	0,326	0,31	0,31	0,29	0,29	0,29	0,29	0,287
0,34	0,34	0,33	0,33	0,33	0,33	0,326	0,31	0,31	0,29	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,33	0,33	0,327	0,31	0,31	0,29	0,29	0,29	0,29	0,287
0,34	0,34	0,33	0,33	0,33	0,33	0,329	0,31	0,31	0,29	0,29	0,29	0,29	0,287
0,34	0,34	0,33	0,33	0,33	0,33	0,329	0,31	0,31	0,3	0,29	0,29	0,29	0,29
0,34	0,34	0,331	0,33	0,33	0,33	0,33	0,31	0,31	0,3	0,29	0,29	0,29	0,29
0,34	0,34	0,332	0,33	0,33	0,33	0,33	0,31	0,31	0,29	0,29	0,29	0,29	0,287
0,34	0,34	0,332	0,33	0,33	0,33	0,33	0,31	0,31	0,29	0,29	0,29	0,29	0,287
0,34	0,34	0,334	0,33	0,33	0,33	0,33	0,31	0,31	0,29	0,29	0,29	0,29	0,287
0,34	0,34	0,334	0,33	0,33	0,33	0,33	0,31	0,31	0,29	0,29	0,29	0,29	0,29
0,34	0,34	0,334	0,33	0,33	0,33	0,33	0,31	0,31	0,3	0,29	0,29	0,29	0,29
0,34	0,34	0,335	0,33	0,33	0,33	0,33	0,31	0,31	0,29	0,29	0,29	0,29	0,287
0,351	0,34	0,34	0,34	0,33	0,33	0,33	0,33	0,32	0,315	0,31	0,297	0,295	0,29
0,36	0,36	0,34	0,34	0,33	0,33	0,31	0,31	0,293	0,29	0,29	0,29	0,29	0,286
0,377	0,34	0,34	0,33	0,33	0,33	0,33	0,31	0,31	0,29	0,29	0,29	0,29	0,287
0,385	0,34	0,34	0,33	0,33	0,33	0,33	0,31	0,31	0,29	0,29	0,29	0,29	0,288

Tabela 33. Estado das 14 piores cargas das soluções relativas à rede 23, para pares de percursos disjuntos nos nós – Fases 1 e 2

0,39	0,39	0,387	0,38	0,38	0,37	0,37	0,37	0,37	0,36	0,36	0,36	0,36
0,39	0,39	0,389	0,37	0,37	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,36
0,39	0,39	0,39	0,39	0,378	0,37	0,37	0,37	0,37	0,36	0,36	0,36	0,36
0,39	0,39	0,39	0,39	0,379	0,37	0,37	0,37	0,37	0,36	0,36	0,36	0,36
0,39	0,39	0,39	0,39	0,38	0,37	0,37	0,37	0,37	0,36	0,36	0,36	0,36
0,39	0,39	0,39	0,39	0,38	0,37	0,37	0,37	0,37	0,36	0,36	0,36	0,36
0,39	0,39	0,39	0,39	0,38	0,38	0,38	0,37	0,37	0,37	0,37	0,36	0,36
0,39	0,39	0,39	0,39	0,381	0,37	0,37	0,37	0,37	0,36	0,36	0,36	0,36
0,39	0,39	0,39	0,39	0,383	0,38	0,38	0,37	0,37	0,37	0,37	0,36	0,36
0,39	0,39	0,39	0,39	0,384	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,36
0,39	0,39	0,39	0,39	0,387	0,38	0,38	0,37	0,37	0,37	0,37	0,36	0,36
0,391	0,39	0,39	0,39	0,39	0,39	0,39	0,37	0,37	0,37	0,37	0,36	0,36
0,391	0,39	0,39	0,39	0,39	0,39	0,39	0,37	0,37	0,37	0,37	0,36	0,36
0,392	0,39	0,39	0,39	0,39	0,39	0,38	0,38	0,374	0,37	0,37	0,36	0,36
0,394	0,39	0,39	0,39	0,39	0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36
0,395	0,39	0,39	0,39	0,39	0,39	0,39	0,39	0,39	0,37	0,37	0,36	0,36
0,397	0,39	0,39	0,39	0,39	0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36
0,398	0,39	0,39	0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,36
0,4	0,39	0,39	0,39	0,39	0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36
0,414	0,41	0,41	0,39	0,39	0,39	0,39	0,37	0,37	0,37	0,37	0,36	0,36
0,425	0,41	0,41	0,39	0,39	0,39	0,39	0,37	0,37	0,365	0,365	0,35	0,34
0,44	0,43	0,425	0,415	0,41	0,4	0,4	0,39	0,39	0,39	0,39	0,384	0,37

Tabela 34. Estado das 13 piores cargas das soluções relativas à rede 23A, para pares de percursos disjuntos nas ligações – Fases 1 e 2

0,39	0,39	0,384	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,36	0,348	0,344
0,39	0,39	0,386	0,37	0,37	0,36	0,36	0,36	0,36	0,348	0,34	0,34	0,34
0,39	0,39	0,387	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,35	0,342	0,34
0,39	0,39	0,389	0,37	0,37	0,36	0,36	0,36	0,36	0,34	0,34	0,34	0,34
0,39	0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,34	0,34	0,34	0,34
0,39	0,39	0,39	0,39	0,375	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,35
0,39	0,39	0,39	0,39	0,376	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,34
0,39	0,39	0,39	0,39	0,376	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,344
0,39	0,39	0,39	0,39	0,377	0,37	0,37	0,36	0,36	0,36	0,36	0,34	0,34
0,39	0,39	0,39	0,39	0,378	0,37	0,37	0,36	0,36	0,36	0,36	0,34	0,34
0,39	0,39	0,39	0,39	0,38	0,37	0,37	0,36	0,36	0,36	0,36	0,34	0,34
0,409	0,39	0,39	0,39	0,37	0,37	0,37	0,37	0,365	0,36	0,36	0,345	0,345
0,42	0,41	0,41	0,394	0,39	0,39	0,39	0,39	0,37	0,37	0,37	0,37	0,35
0,431	0,39	0,39	0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,34	0,34
0,432	0,39	0,39	0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,34	0,34
0,432	0,39	0,39	0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,34	0,34
0,435	0,39	0,39	0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,34	0,34
0,438	0,39	0,39	0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,34	0,34
0,439	0,39	0,39	0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,34	0,34
0,439	0,39	0,39	0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,34	0,34
0,439	0,39	0,39	0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,34	0,34
0,442	0,39	0,39	0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,34	0,34

Tabela 35. Estado das 13 piores cargas das soluções relativas à rede 23A, para pares de percursos disjuntos nos nós – Fases 1 e 2



[illegible]

Tabela 38. Estado das 14 piores cargas das soluções relativas à rede 29A, para pares de percursos disjuntos nas ligações – Fases 1 e 2

[illegible]

Tabela 39. Estado das 14 piores cargas das soluções relativas à rede 29A, para pares de percursos disjuntos nos nós – Fases 1 e 2

0,34	0,34	0,33	0,33	0,31	0,31	0,29	0,29	0,29	0,29	0,29	0,29	0,285
0,34	0,34	0,33	0,33	0,31	0,31	0,291	0,29	0,29	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,292	0,29	0,29	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,292	0,292	0,292	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,293	0,29	0,29	0,29	0,29	0,29	0,286
0,34	0,34	0,33	0,33	0,31	0,31	0,3	0,293	0,292	0,292	0,29	0,29	0,28
0,34	0,34	0,33	0,33	0,31	0,31	0,3	0,293	0,292	0,292	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,3	0,3	0,295	0,292	0,292	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,3	0,3	0,297	0,292	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,302	0,295	0,29	0,29	0,29	0,29	0,285
0,34	0,34	0,33	0,33	0,31	0,31	0,303	0,3	0,3	0,292	0,29	0,29	0,289
0,34	0,34	0,33	0,33	0,31	0,31	0,305	0,305	0,297	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,31	0,3	0,294	0,293	0,292	0,292	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,31	0,305	0,305	0,3	0,292	0,29	0,29
0,34	0,34	0,33	0,33	0,314	0,312	0,31	0,31	0,31	0,31	0,304	0,298	0,29
0,34	0,34	0,33	0,33	0,315	0,314	0,314	0,314	0,31	0,31	0,307	0,29	0,29
0,34	0,34	0,33	0,33	0,319	0,316	0,31	0,31	0,308	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,32	0,312	0,31	0,305	0,305	0,301	0,3	0,29	0,29
0,34	0,34	0,33	0,33	0,322	0,31	0,31	0,31	0,305	0,3	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,329	0,322	0,322	0,31	0,31	0,3	0,292	0,29	0,29
0,34	0,34	0,33	0,33	0,33	0,326	0,32	0,32	0,32	0,31	0,31	0,31	0,3
0,34	0,34	0,34	0,332	0,33	0,33	0,33	0,322	0,32	0,32	0,31	0,31	0,31
0,34	0,34	0,34	0,34	0,336	0,33	0,33	0,32	0,31	0,31	0,31	0,31	0,31
0,343	0,34	0,34	0,33	0,33	0,33	0,315	0,31	0,31	0,307	0,29	0,29	0,29
0,35	0,34	0,34	0,334	0,33	0,33	0,33	0,325	0,32	0,31	0,31	0,31	0,3
0,35	0,34	0,34	0,34	0,34	0,33	0,33	0,33	0,31	0,31	0,31	0,3	0,292
0,35	0,34	0,34	0,34	0,34	0,34	0,33	0,33	0,31	0,31	0,31	0,29	0,29
0,35	0,348	0,342	0,34	0,34	0,33	0,33	0,31	0,31	0,31	0,3	0,292	0,29
0,35	0,349	0,342	0,34	0,34	0,34	0,33	0,33	0,31	0,31	0,3	0,3	0,292
0,351	0,34	0,34	0,33	0,33	0,33	0,32	0,32	0,31	0,31	0,31	0,3	0,292

Tabela 40. Estado das 13 piores cargas das soluções relativas à rede 23, para pares de percursos disjuntos nas ligações – Fase 3

0,34	0,34	0,33	0,33	0,31	0,31	0,295	0,295	0,294	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,295	0,295	0,294	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,297	0,294	0,293	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,297	0,294	0,293	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,297	0,296	0,291	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,297	0,296	0,291	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,297	0,296	0,291	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,297	0,297	0,29	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,298	0,296	0,29	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,299	0,296	0,29	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,299	0,298	0,29	0,29	0,29	0,29	0,287
0,34	0,34	0,33	0,33	0,31	0,31	0,3	0,296	0,294	0,294	0,292	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,3	0,296	0,295	0,293	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,3	0,297	0,297	0,292	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,3	0,3	0,294	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,3	0,3	0,3	0,3	0,298	0,292	0,292

0,34	0,34	0,33	0,33	0,31	0,31	0,304	0,301	0,3	0,292	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,305	0,302	0,299	0,29	0,29	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,31	0,302	0,3	0,295	0,294	0,29	0,29
0,34	0,34	0,33	0,33	0,31	0,31	0,31	0,306	0,3	0,297	0,292	0,29	0,29
0,34	0,34	0,33	0,33	0,318	0,31	0,31	0,31	0,3	0,3	0,292	0,29	0,29
0,34	0,34	0,332	0,33	0,33	0,32	0,31	0,31	0,31	0,31	0,3	0,3	0,297
0,34	0,34	0,335	0,33	0,33	0,329	0,32	0,31	0,31	0,31	0,31	0,296	0,29
0,34	0,34	0,34	0,33	0,33	0,327	0,32	0,315	0,31	0,31	0,31	0,307	0,3
0,34	0,34	0,34	0,33	0,33	0,33	0,329	0,32	0,32	0,315	0,31	0,297	0,297
0,34	0,34	0,34	0,34	0,33	0,33	0,33	0,33	0,32	0,315	0,31	0,3	0,295
0,342	0,34	0,34	0,33	0,33	0,33	0,32	0,316	0,31	0,31	0,31	0,3	0,295
0,346	0,34	0,34	0,34	0,33	0,33	0,33	0,33	0,32	0,315	0,31	0,297	0,295
0,347	0,34	0,34	0,34	0,33	0,33	0,33	0,33	0,32	0,315	0,297	0,295	0,29
0,351	0,34	0,34	0,33	0,33	0,33	0,33	0,31	0,31	0,31	0,31	0,31	0,297

Tabela 41. Estado das 13 piores cargas das soluções relativas à rede 23, para pares de percursos disjuntos nos nós – Fase 3

0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,342	0,34	0,34	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,344	0,34	0,34	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,344	0,34	0,34	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,348	0,34	0,34	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,35	0,343	0,342	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,36	0,34	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,36	0,343	0,342	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,36	0,35	0,343	0,342
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,36	0,352	0,352	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,36	0,353	0,35	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,36	0,359	0,34	0,34
0,39	0,39	0,37	0,37	0,362	0,36	0,36	0,36	0,36	0,36	0,356	0,356	0,344
0,39	0,39	0,37	0,37	0,364	0,364	0,36	0,36	0,36	0,36	0,35	0,35	0,342
0,39	0,39	0,37	0,37	0,37	0,362	0,36	0,36	0,354	0,35	0,342	0,34	0,34
0,39	0,39	0,37	0,37	0,37	0,362	0,36	0,36	0,36	0,35	0,344	0,342	0,342
0,39	0,39	0,37	0,37	0,37	0,362	0,36	0,36	0,36	0,352	0,35	0,35	0,34
0,39	0,39	0,37	0,37	0,37	0,368	0,362	0,36	0,36	0,36	0,36	0,354	0,346
0,39	0,39	0,38	0,37	0,37	0,364	0,362	0,36	0,36	0,36	0,36	0,359	0,354
0,39	0,39	0,38	0,38	0,37	0,37	0,367	0,36	0,36	0,36	0,36	0,35	0,342
0,39	0,39	0,39	0,39	0,388	0,38	0,37	0,37	0,37	0,365	0,365	0,345	0,34
0,392	0,392	0,39	0,39	0,39	0,386	0,38	0,372	0,365	0,36	0,36	0,36	0,344
0,395	0,395	0,39	0,39	0,38	0,38	0,379	0,37	0,37	0,36	0,36	0,36	0,35
0,395	0,395	0,39	0,39	0,39	0,39	0,37	0,37	0,37	0,365	0,345	0,34	0,34
0,405	0,4	0,397	0,395	0,392	0,39	0,39	0,38	0,37	0,37	0,36	0,36	0,34
0,41	0,41	0,39	0,39	0,39	0,377	0,372	0,37	0,37	0,365	0,36	0,355	0,34
0,41	0,41	0,39	0,39	0,39	0,39	0,37	0,37	0,37	0,368	0,365	0,365	0,34
0,41	0,41	0,391	0,39	0,39	0,39	0,39	0,38	0,37	0,37	0,365	0,365	0,34
0,41	0,41	0,393	0,39	0,39	0,39	0,37	0,37	0,37	0,365	0,362	0,35	0,35
0,41	0,41	0,406	0,404	0,39	0,39	0,387	0,37	0,37	0,37	0,36	0,36	0,36
0,42	0,395	0,392	0,39	0,39	0,38	0,38	0,375	0,375	0,37	0,37	0,36	0,36

Tabela 42. Estado das 13 piores cargas das soluções relativas à rede 23A, para pares de percursos disjuntos nas ligações – Fase 3

0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,345	0,345	0,344	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,346	0,344	0,344	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,346	0,344	0,344	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,346	0,345	0,343	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,346	0,346	0,342	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,347	0,346	0,341	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,347	0,347	0,34	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,348	0,344	0,342	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,349	0,346	0,34	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,349	0,348	0,34	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,35	0,345	0,345	0,344	0,342
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,35	0,346	0,346	0,342	0,342
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,35	0,347	0,34	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,35	0,35	0,346	0,342	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,35	0,35	0,35	0,347	0,342
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,351	0,342	0,341	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,352	0,35	0,346	0,342	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,352	0,352	0,349	0,348	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,354	0,34	0,34	0,34	0,34
0,39	0,39	0,37	0,37	0,36	0,36	0,36	0,36	0,36	0,36	0,35	0,35	0,348
0,39	0,39	0,37	0,37	0,37	0,362	0,36	0,36	0,352	0,352	0,345	0,345	0,344
0,39	0,39	0,37	0,37	0,37	0,362	0,362	0,36	0,36	0,354	0,354	0,341	0,34
0,39	0,39	0,386	0,385	0,38	0,377	0,37	0,37	0,37	0,362	0,36	0,36	0,349
0,39	0,39	0,39	0,382	0,38	0,38	0,38	0,38	0,38	0,37	0,37	0,37	0,365
0,39	0,39	0,39	0,385	0,38	0,38	0,377	0,37	0,37	0,365	0,361	0,36	0,36
0,39	0,39	0,39	0,39	0,375	0,372	0,37	0,37	0,37	0,365	0,365	0,355	0,35
0,39	0,39	0,39	0,39	0,38	0,37	0,37	0,37	0,365	0,36	0,36	0,345	0,34
0,397	0,39	0,39	0,39	0,38	0,37	0,37	0,37	0,365	0,36	0,36	0,348	0,345
0,397	0,39	0,39	0,39	0,39	0,38	0,37	0,37	0,37	0,365	0,365	0,35	0,348
0,4	0,39	0,39	0,39	0,39	0,39	0,39	0,37	0,37	0,37	0,365	0,36	0,36

Tabela 43. Estado das 13 piores cargas das soluções relativas à rede 23A, para pares de percursos disjuntos nos nós – Fase 3

[illegible]





[illegible][illegible]



# Bibliografia

- [1] Beethovem Zanella Dias e Nilton Alves Jr., “Evolução do Padrão Ethernet”, 2002
- [2] LAN SWITCHING - ETHERNET
  - . [http://www.cisco.com/en/US/tech/tk389/tk214/tsd\\_technology\\_support\\_protocol\\_home.html](http://www.cisco.com/en/US/tech/tk389/tk214/tsd_technology_support_protocol_home.html)
- [3] Ethernet
  - . <http://en.wikipedia.org/wiki/Ethernet>
- [4] Nortel, “Provider Backbone Bridges bring massive service scalability to Ethernet”, 2007
- [5] Provider Backbone Bridges
  - . [http://en.wikipedia.org/wiki/Provider\\_Backbone\\_Bridges](http://en.wikipedia.org/wiki/Provider_Backbone_Bridges)
- [6] Provider Backbone Bridge - Traffic Engineering
  - . [http://www.nortel.com/corporate/technology\\_new/technologies.html](http://www.nortel.com/corporate/technology_new/technologies.html)
- [7] TPACK, “PBB-TE, PBT Carrier Grade Ethernet Transport”, 2007
- [8] Provider Backbone Bridge - Traffic Engineering
  - . [http://en.wikipedia.org/wiki/IEEE\\_802.1Qay](http://en.wikipedia.org/wiki/IEEE_802.1Qay)
- [9] Open Systems Interconnection (OSI) Protocols
  - . [http://www.cisco.com/en/US/tech/tk389/tk214/tsd\\_technology\\_support\\_protocol\\_home.html](http://www.cisco.com/en/US/tech/tk389/tk214/tsd_technology_support_protocol_home.html)
- [9] Open Systems Interconnection (OSI) Protocols
  - . [http://www.cisco.com/en/US/tech/tk389/tk214/tsd\\_technology\\_support\\_protocol\\_home.html](http://www.cisco.com/en/US/tech/tk389/tk214/tsd_technology_support_protocol_home.html)
- [10] Intel and Ethernet
  - . [http://www.intel.com/standards/case/case\\_ethernet.htm](http://www.intel.com/standards/case/case_ethernet.htm)
- [11] Lawrence Harte, “Introduction to Data Networks”, Althos Publishing, 2006
- [12] Apontamentos de Redes de Telecomunicações, “Protocolos de Acesso ao Meio / Redes Ethernet”
- [13] IEEE 802.11 LAN/MAN Wireless LANS
  - . <http://standards.ieee.org/getieee802/802.11.html>
- [14] Data Link Layer
  - . [http://en.wikipedia.org/wiki/Data\\_Link\\_Layer](http://en.wikipedia.org/wiki/Data_Link_Layer)
- [15] Nortel
  - . <http://www.nortel.com/>
- [16] Fujitsu, “Understanding PBB-TE for Carrier Ethernet”, 2008
- [17] Ramesh Bhandari, “Survivable Networks: Algorithms for Diverse Routing”, Kluwer Academic Publishers, 1999
- [18] A. F. de Sousa, “Algoritmos Eficientes para Problemas de Grafos”, 2006
- [19] E.W. Dijkstra, “A note on two problems in connexion with graphs”, 269–271, Numerische

Mathematik 1, 1959

[20] F. Benjamin Zhan, "Unit 064 - Representing Networks", 1998

. <http://www.ncgia.ucsb.edu/giscc/units/u064/>